



PUBLIC

Document Version: 1H 2024 – 2024-04-19

# Using the Employee Delta Export Add-In for Microsoft Excel

# Content

<b>1</b>	<b>Before You Start. . . . .</b>	<b>4</b>
<b>2</b>	<b>Introduction to Employee Delta Export for Microsoft Excel. . . . .</b>	<b>5</b>
2.1	Feature Overview of the Employee Delta Export for Microsoft Excel. . . . .	5
2.2	Target Group of This Document. . . . .	6
<b>3</b>	<b>Setting Up Employee Delta Export for Microsoft Excel. . . . .</b>	<b>7</b>
3.1	Technical Prerequisites for Using the Employee Delta Export for Microsoft Excel. . . . .	7
3.2	Special Handling for the Employee Delta Export for Microsoft Excel. . . . .	8
	Behaviors Caused by the Compound Employee API. . . . .	9
	Behaviors Caused by the Client Application. . . . .	10
3.3	Security and Session Management in the Employee Delta Export for Microsoft Excel. . . . .	11
	Setting up the OAuth 2.0 Application. . . . .	11
	Restricting User Account for API Calls to Certain IP Addresses. . . . .	13
	Setting Role-Based Permissions for the Employee Delta Export Add-In for Microsoft Excel. . . . .	13
3.4	Application Components of the Employee Delta Export for Microsoft Excel. . . . .	14
3.5	Employee Delta Export UI. . . . .	15
3.6	Version Check and Updates. . . . .	15
3.7	Installing the Employee Delta Export Add-In for Microsoft Excel. . . . .	16
3.8	Uninstalling the Employee Delta Export Add-In for Microsoft Excel. . . . .	16
<b>4</b>	<b>Working with Employee Delta Export for Microsoft Excel. . . . .</b>	<b>17</b>
4.1	First Steps to Using the Employee Delta Export for Microsoft Excel. . . . .	17
	API Endpoint URLs for Employee Delta Export. . . . .	18
	Carrying Out an Example Test Run. . . . .	19
4.2	Architecture Concept of the Employee Delta Export for Microsoft Excel. . . . .	19
	Component Structure. . . . .	19
	Data Extraction Process. . . . .	20
	How Delta Calculation Works. . . . .	20
4.3	Workbooks for Employee Delta Export for Microsoft Excel. . . . .	28
4.4	Worksheets of Employee Delta Export for Microsoft Excel. . . . .	29
	Result Sheets. . . . .	30
	Selection Sheet. . . . .	34
	Summary Sheets. . . . .	42
	Systems Sheet. . . . .	42
	Config Sheet. . . . .	43
	Errors Sheet. . . . .	46
	Warnings Sheet. . . . .	47

	Internal Sheets. . . . .	47
4.5	Tabs on the Ribbon. . . . .	48
	SuccessFactors Tab. . . . .	49
	Developer Tab. . . . .	49
<b>5</b>	<b>Extending Employee Delta Export for Microsoft Excel. . . . .</b>	<b>57</b>
5.1	What Types of Changes Can Be Applied to Reporting Workbooks?. . . . .	58
5.2	How Column Configuration Works. . . . .	59
5.3	Adding New Columns from the Database. . . . .	60
5.4	Adding Calculated Columns. . . . .	61
5.5	Adding User-Friendly Descriptions for Picklist-Based Columns. . . . .	62
5.6	Adding User-Friendly Descriptions for Foundation Object-Based Columns. . . . .	63
5.7	Deleting Columns. . . . .	64
5.8	Adding Personal Data to User ID Fields. . . . .	65
5.9	Adding Information About Workflow Approvals. . . . .	66
5.10	Adding New Filter Parameters. . . . .	67
5.11	Using Payment Information in the Result Sheets. . . . .	68
5.12	Choosing Which Segments Are Extracted. . . . .	69
5.13	Dos and Don'ts for Using the Employee Delta Export Add-In for Microsoft Excel. . . . .	70
<b>6</b>	<b>Troubleshooting Employee Delta Export to Microsoft Excel. . . . .</b>	<b>72</b>

# 1 Before You Start

What you need to know before you start using the Employee Delta Export Add-In for Microsoft Excel.

The application described in this document **isn't** subject to your license agreement or any other agreement with SAP SuccessFactors.

SAP SuccessFactors has no obligation to pursue any course of business outlined in this document or to develop or release any functionality mentioned in this document. This document and SAP SuccessFactors' strategy and possible future developments are subject to change and may be changed by SAP SuccessFactors at any time for any reason without notice. This document is provided without a warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or noninfringement. SAP SuccessFactors assumes no responsibility for errors or omissions in this document.

## Note

The Employee Delta Export Add-In for Microsoft Excel is on maintenance mode only. No development of new features is planned.

## 2 Introduction to Employee Delta Export for Microsoft Excel

Employee Delta Export for Microsoft Excel is an Excel-based application that retrieves employee master data from Employee Central, which can then be used for payroll runs in a subsequent system.

The Employee Delta Export provides new employee master data and employee master data that was changed compared to the end time and date of the previous run from Employee Central. The export minimizes the effort of transferring the changed employee data to the payroll system. It considers changes on all time slices of effective-dated employee data. In other words, it considers future-dated and retroactive changes of employee data. It indicates data deletions in certain cases, for example, if a pay component was deleted.

The Employee Delta Export application consists of an add-in for Microsoft Excel and two workbooks. The workbooks provide the layout of the data being retrieved and some configuration data. The configuration data allows for consultants and administrators to configure the data retrieval process without actually having to program.

### Note

If you want to use Employee Delta Export for Microsoft Excel, you need advanced knowledge of working with Microsoft Excel and XML.

[Feature Overview of the Employee Delta Export for Microsoft Excel \[page 5\]](#)

Take a look at the main functions of Employee Delta Export for Microsoft Excel.

[Target Group of This Document \[page 6\]](#)

This document is intended for consultants, partners, and customers who would like to learn more about the Employee Delta Export for Microsoft Excel, its technical background, and how to use it.

## Related Information

[How Delta Calculation Works \[page 20\]](#)

### 2.1 Feature Overview of the Employee Delta Export for Microsoft Excel

Take a look at the main functions of Employee Delta Export for Microsoft Excel.

The most important features provided by the Employee Delta Export Add-In for Microsoft Excel are:

- Filter employees who have been changed, hired, or re-hired since the begin of the evaluation period
- Show before image and current image of changed data

- Consider changes along the full history of employees (retroactive changes and future-dated changes)
- Highlight changes for easy manual processing
- Support all selection criteria offered by the underlying API in the backend
- Support additional selection criteria that are evaluated in Excel locally, on your machine
- Lookup user-friendly descriptions for picklist and foundation object based fields
- Lookup additional information for related persons (for example manager's first/last name)
- Include workflow approval information
- Easy configuration of columns shown on the worksheets without programming
- Easy configuration of how the returned changes on each sheet are sorted
- Support for locally calculated columns derived from backend fields
- User-defined selection criteria
- User-defined table styles
- One click installation of the tool
- Passwords are not persisted
- Trouble-shooting support
- Version check for the add-in and workbook files that are available in the [Employee Delta Export UI](#)
- Request certain segments to be extracted and processed

## 2.2 Target Group of This Document

This document is intended for consultants, partners, and customers who would like to learn more about the Employee Delta Export for Microsoft Excel, its technical background, and how to use it.

# 3 Setting Up Employee Delta Export for Microsoft Excel

What needs to be set up before you can start using the Employee Delta Export Add-In for Microsoft Excel to extract changes made to employee master data in Employee Central.

## [Technical Prerequisites for Using the Employee Delta Export for Microsoft Excel \[page 7\]](#)

Your system landscape needs to fulfill some prerequisites if you want to use Employee Delta Export for Microsoft Excel.

## [Special Handling for the Employee Delta Export for Microsoft Excel \[page 8\]](#)

Employee Delta Export for Microsoft Excel has some specific behaviors that are caused either by the Compound Employee API or by the client application.

## [Security and Session Management in the Employee Delta Export for Microsoft Excel \[page 11\]](#)

The Employee Delta Export for Microsoft Excel requires logon, logout, session management, and role-based-permissions.

## [Application Components of the Employee Delta Export for Microsoft Excel \[page 14\]](#)

The Employee Delta Export for Microsoft Excel consists of an Excel add-in and one or more reporting workbooks.

## [Employee Delta Export UI \[page 15\]](#)

On the *Employee Delta Export* UI you can find the Microsoft Excel add-in installer, the sample workbooks, and this user guide for download.

## [Version Check and Updates \[page 15\]](#)

The *Employee Delta Export* UI ensures that you are using the newest version of the Employee Delta Export Add-In for Microsoft Excel.

## [Installing the Employee Delta Export Add-In for Microsoft Excel \[page 16\]](#)

Install the Employee Delta Export Add-In for Microsoft Excel on your local computer.

## [Uninstalling the Employee Delta Export Add-In for Microsoft Excel \[page 16\]](#)

Uninstall the Employee Delta Export Add-In for Microsoft Excel from your local computer if you no longer need it.

## 3.1 Technical Prerequisites for Using the Employee Delta Export for Microsoft Excel

Your system landscape needs to fulfill some prerequisites if you want to use Employee Delta Export for Microsoft Excel.

- One of the following Microsoft Excel for Windows versions is required:
  - Microsoft Excel 2021
  - Microsoft 365 Excel
- Windows version 10 or higher is required.

- Your company policy must allow the execution of macros in Microsoft Excel.
- Microsoft XML Core Services version 6 (MSXML 6.0) or newer must be installed on the computer.
- SAP SuccessFactors Employee Central must be up and running, and the following settings must be in place:
  - [Employee Profile Data Audit](#) must be turned on in Provisioning, under [Company Settings](#). For more information, refer to *Setting Up the Basics* in the *Implementing Employee Central Core* guide in SAP Help Portal.
  - The `CompoundEmployee` API must be enabled. For more information, see *Setting Up the Web Service* in the *Implementing Employee Central Payroll* guide.
  - OData APIs must be enabled.
  - MDF PicklistV2 must be enabled in your instance.
  - All required permissions to use OData APIs and access MDF PicklistV2 entities are set up.
  - The following field names of the [Lookup\\_Person](#) sheet are new. They're automatically provided by downloading the latest version of the tool or by replacing them manually:
    - `first_name` with `firstName`
    - `last_name` with `lastName`
    - `middle_name` with `middleName`
  - The OAuth 2.0 Client Application has to be established to use the Employee Delta Export for Microsoft Excel.

## Related Information

[Setting Up the Basics](#)

[Setting Up Web Service](#)

[Setting up the OAuth 2.0 Application \[page 11\]](#)

## 3.2 Special Handling for the Employee Delta Export for Microsoft Excel

Employee Delta Export for Microsoft Excel has some specific behaviors that are caused either by the `CompoundEmployee` API or by the client application.

[Behaviors Caused by the Compound Employee API \[page 9\]](#)

Consider these behaviors if you use the Employee Delta Export for Microsoft Excel.

[Behaviors Caused by the Client Application \[page 10\]](#)

When using the Employee Delta Export Add-In for Microsoft Excel, consider these behaviors caused by the client system.



## 3.2.1 Behaviors Caused by the Compound Employee API

Consider these behaviors if you use the Employee Delta Export for Microsoft Excel.

- Role-based permissions aren't supported. This means you need to grant the permission [Employee Central HRIS SOAP API](#) to the user who uses the Employee Delta Export. This allows the user to retrieve all kinds of data that is accessible using SOAP APIs.
- The Employee Delta Export can only handle the following entities:
  - Person
  - Personal Information
  - Personal Information Global
  - Address Information
  - Employment Information
  - Job Information
  - Compensation Information
  - Pay Compensation Recurring
  - Pay Compensation Non-Recurring
  - PaymentInformationV3
  - National ID Card
  - Email
  - Phone
  - Person Relation
  - Job Relation
  - Global Assignment Information

The following segments aren't included in the standard query. You can request to have these segments included as result sheets.

- Deduction Recurring
- One Time Deduction
- Alternative Cost Distribution
- Personal Documents Information
- The Employee Delta Export doesn't consider an employee changed when other entities than the above have changed (for example, foundation objects used for lookup purposes)
- By default, the workbook shows all future time slices of an entity that have been changed, meaning changes that become effective after the payroll period can't be excluded. However, if you use [Period-based Delta](#), you can determine the start and end date of the validity period for which you want time slices to be considered.
- Since the Employee Delta Export is primarily based on the `CompoundEmployee` API, it inherits most of the limitations of the `CompoundEmployee` API that are:
  - Event and event reason on Compensation Information isn't supported
  - Only filter parameters supported by the `CompoundEmployee` API are passed on to the API server. Other filters are applied locally in Microsoft Excel. This may lead to unexpected high data volumes being retrieved from the API server.
  - Picklists and Foundation Objects assigned to fields returned by the API need to have external codes. Otherwise, no value is returned for those fields.
  - Severe data inconsistencies (for example time slice mismatches) may prevent employees from being shown. They appear on the [Errors](#) sheet instead.

## Related Information

[Using Payment Information in the Result Sheets \[page 68\]](#)

[Choosing Which Segments Are Extracted \[page 69\]](#)

[How Period-Based Delta Works \[page 40\]](#)

### 3.2.2 Behaviors Caused by the Client Application

When using the Employee Delta Export Add-In for Microsoft Excel, consider these behaviors caused by the client system.

- For performance reasons, the Employee Delta Export shouldn't be used to process more than 1000 employees. Depending on the data volume of the processed employees, the maximum number of processed employees can be substantially smaller. In such cases, the data needs to be broken down into smaller parts that can be handled by Microsoft Excel.
- Initial load scenarios that extract the whole change history for every employee of a target population aren't supported. A delta extraction must have a report start date.
- It's not possible to specify country/region-specific segments to be extracted using the Employee Delta Export Add-In.
- No localization is provided for:
  - Workbook static content (text, messages)
  - Employee data (descriptions)
  - Foundation data (for example, pay component names)
  - Picklist entries (for example, *married*, *single*, *divorced*)
- When you plan to divide your overall employee population into different pieces processed by different processors using different filter criteria, this needs to be planned thoroughly to prevent loss of information when employees transfer from one subpopulation to another.
- You can't use lists of values for filter parameters in arbitrary combinations.
- Don't hide any rows. Data in hidden rows might interfere with new data being written to the Microsoft Excel sheets.
- Sorting must be done locally in Microsoft Excel.
- Single sign-on isn't supported.
- The Employee Delta Export makes it possible to manually extract data from a system. Even though it's possible to have an automated process read data from the processed workbook, the Employee Delta Export doesn't provide a functionality to automatically start the export.
- Contingent Worker employments aren't supported by Employee Delta Export. Since an employee that is a contingent worker has only a contingent worker employment, that employee isn't considered by the Employee Delta Export as a whole.

## 3.3 Security and Session Management in the Employee Delta Export for Microsoft Excel

The Employee Delta Export for Microsoft Excel requires logon, logout, session management, and role-based-permissions.

The Employee Delta Export for Microsoft Excel calls the `CompoundEmployee` API to extract data from the system. When you trigger the extraction, a logon dialog pops up asking for the following OAuth 2.0 keys from the selected system:

- [User ID](#)
- [API Key](#)
- encrypted [Private Key](#) from the X.509 Certificate

The OAuth 2.0 application is established through the logon operation using the binding [User ID](#), [API Key](#) and an encrypted [Private Key](#).

### [Setting up the OAuth 2.0 Application \[page 11\]](#)

Learn how to set up the OAuth 2.0 Client Application.

### [Restricting User Account for API Calls to Certain IP Addresses \[page 13\]](#)

Restrict a user account used for API access – including the API access used by Employee Delta Export for Microsoft Excel – to certain IP addresses.

### [Setting Role-Based Permissions for the Employee Delta Export Add-In for Microsoft Excel \[page 13\]](#)

Set the role-based permissions required to use Employee Delta Export for Microsoft Excel.

## Related Information

[Setting up the OAuth 2.0 Application \[page 11\]](#)

### 3.3.1 Setting up the OAuth 2.0 Application

Learn how to set up the OAuth 2.0 Client Application.

## Prerequisites

You can set up the OAuth 2.0 Client Application for the Employee Delta Export for Microsoft Excel Logon from the *Authentication Using OAuth 2.0* page in the [Related Information](#) section.

#### Note

[Bind to Users](#) is mandatory for the OAuth 2.0 Logon process in the Employee Delta Export for Microsoft Excel due to security reasons.

For the registration of the OAuth 2.0 Client Application, the [Application Name](#) and the [Application URL](#) must be unique within one company. Therefore, for the [Application URL](#) of the OAuth 2.0 Client Application, you must use a unique URL. For example, this could be a link with a reference to information specific to the binding user.

### ❖ Example

An example of a configuration in [Manage OAuth 2.0 Client Applications](#) appears as follows:

Application Name = ExcelOAuth2.0 for <userId>

Application URL = https://corporate-addressbook.local/<userId>

Bind to Users = yes

User ID = <userId>

## Procedure

1. After the successful creation of the OAuth 2.0 Client Application, you can begin using the Employee Delta Export for Microsoft Excel.
2. When you select [Get Data](#) on the SuccessFactors tab of the Excel Add-In, an OAuth 2.0 Logon dialog will appear. There you must enter the following OAuth2.0 keys:

[User ID](#): Enter the User ID that you used during the [Bind to Users](#) part of the OAuth 2.0 set up. The [User ID](#) will be stored in the background of the Employee Delta Export for Microsoft Excel Add-In after the first successful logon and remains the same until you close the workbook.

[API Key](#): Enter the API Key that you received when setting up your instance for OAuth2.0. The [API Key](#) will be stored in the background of the Employee Delta Export for Microsoft Excel Add-In after the first successful logon and remains the same until you close the workbook.

[Private Key](#): You must use the [ENCRYPTED PRIVATE KEY](#) from the X.509 Certificate which you downloaded directly after the Setup for the OAuth 2.0. The [Private Key](#) isn't stored in the background for security reasons, and you have to enter it at every logon.

## Related Information

[Authentication Using OAuth 2.0](#)

## 3.3.2 Restricting User Account for API Calls to Certain IP Addresses

Restrict a user account used for API access – including the API access used by Employee Delta Export for Microsoft Excel – to certain IP addresses.

### Procedure

1. Go to the Admin Center and choose the [Password & Login Policy Settings](#) admin tool
2. Choose the link [Set API login exceptions...](#)
3. Choose [Add](#).
4. Enter the following data:
  - [Username](#): The user account used for API access
  - [Maximum password age \(days\)](#): 0
  - [IP address restrictions](#): Your IP addresses

## 3.3.3 Setting Role-Based Permissions for the Employee Delta Export Add-In for Microsoft Excel

Set the role-based permissions required to use Employee Delta Export for Microsoft Excel.

### Context

Grant your users of Employee Delta Export for Microsoft Excel the permissions required for using the CompoundEmployee API and for accessing the [Employee Delta Export](#) UI.

### Procedure

1. Set the permissions required to extract data from the system using the CompoundEmployee API.

Category	Permission
Under <a href="#">User Permissions</a> , select the <a href="#">General User Permission</a> category.	Select the <a href="#">SFAPI User Login</a> permission.
Under <a href="#">Administrator Permissions</a> , select the <a href="#">Employee Central API</a> category.	Select the following permissions: <ul style="list-style-type: none"><li>• <a href="#">Employee Central HRIS SOAP API</a></li><li>• <a href="#">Employee Central Foundation OData API (read-only)</a></li></ul>

Category	Permission
Under <i>Administrator Permissions</i> , select the <i>Manage Integration Tools</i> category.	Select the <i>Allow Admin to Access OData API through Basic Authentication</i> permission.
Under <i>Administrator Permissions</i> , select the <i>Metadata Framework</i> category.	Select the <i>Admin Access to MDF OData API</i> permission.
Under <i>User Permissions</i> , select the <i>Miscellaneous Permissions</i> category.	Search for the <i>Object Picklist</i> and select <i>View Current</i> .
<div> <div> Note </div> <div> This is optional and is only needed in cases where the MDF Object Picklist is set to <i>Secured</i> in the object definition. </div> </div>	

- Set the permission required to access the *Employee Delta Export* UI.

Category	Permission
Under <i>Administrator Permissions</i> , select either the <i>Manage Dashboards / Reports</i> or the <i>Manage User</i> category.	<p>Select the <i>Manage Employee Delta Export Templates</i> permission.</p> <p>If the user logs out and then logs in again, they can now find <i>Employee Delta Export</i> link in Admin Center, using the <i>Admin Tools</i> search field.</p>

## Related Information

[List of Role-Based Permissions](#)

## 3.4 Application Components of the Employee Delta Export for Microsoft Excel

The Employee Delta Export for Microsoft Excel consists of an Excel add-in and one or more reporting workbooks.

The Excel add-in contains the logic to retrieve the data from the backend and perform delta calculation. The add-in adds functions to the Excel ribbon that allow you to fill one of the reporting workbooks with data fetched from the backend.

You can have an arbitrary number of reporting workbooks. The workbook provides the layout for the data to be shown plus configuration information used by the add-in.

## 3.5 Employee Delta Export UI

On the [Employee Delta Export](#) UI you can find the Microsoft Excel add-in installer, the sample workbooks, and this user guide for download.

Find the [Employee Delta Export](#) UI in the Admin Tools, under [Employee Files](#) or [Reporting](#).

### Note

You need the [Manage Employee Delta Export Templates](#) permission to access the [Employee Delta Export](#) UI.

The [Employee Delta Export](#) UI has a list of files you can download:

- [Microsoft Excel Add-In Installer](#) (CompEmpDeltaAddIn.xlsm): Use this to install the add-in.
- [Multi-Sheet Workbook](#) (CompEmpDelta\_multiSheet.xlsx): Contains separate worksheets for all Employee Central entities.
- [Single-Sheet Workbook](#) (CompEmpDelta\_singleSheet.xlsx): A sample workbook showing all Employee Central entities on one single worksheet.
- [User Guide](#) (Employee Delta Excel.pdf): The current *Employee Central Compound Employee API - Employee Delta Export Add-in for Microsoft Excel* guide.

Each file has its own version consisting of three numbers. The first number indicates the SAP SuccessFactors build for which the file is provided. Only the files relevant to your SAP SuccessFactors release are listed. The second number represents the patch version of the file. The add-in performs a version check each time you retrieve data from the Microsoft Excel and give a warning if the add-in or a workbook is outdated. The third number indicates the revision of the file and represents minor changes. You are not forced to update files that have an outdated revision number.

## 3.6 Version Check and Updates

The [Employee Delta Export](#) UI ensures that you are using the newest version of the Employee Delta Export Add-In for Microsoft Excel.

Whenever you execute a query request by selecting [Get Data](#), a version check is performed. If there is a version of the add-in with a higher release number or patch version than the file you have installed, you will get this warning: *Your version is outdated and may cause problems. We strongly recommend you download the latest version of the Employee Delta Export Add-in from Admin Tools.*

If only the revision number of your installed add-in differs from that of the newest version available, you get a different warning: *A new version of the Employee Delta Export Add-in is available. Please download it from Admin Tools to benefit from the latest features. You can continue working in the older version and download a new version later, if desired.*

If you select the [Do not show this message again](#) checkbox, the warning won't show up again as long as the revision number of the add-in file in the [Employee Delta Export](#) UI isn't changed. Revision changes are minor changes that won't influence existing features. Therefore, updating is optional.

The compatibility of the add-in and the workbook is also checked.

## 3.7 Installing the Employee Delta Export Add-In for Microsoft Excel

Install the Employee Delta Export Add-In for Microsoft Excel on your local computer.

### Procedure

1. Go to the Admin Center and choose the [Employee Delta Export](#) admin tool.
2. Download [Excel Add-In Installer](#) (CompEmpDeltaAddIn.xlsm).
3. After downloading the installer, open it and follow the instructions shown in the pop-up.

To avoid security issues, ensure that macro execution is enabled in Microsoft Excel and that the workbook is made a trusted document. The software is digitally signed by SAP to ensure corporate compliance.

## 3.8 Uninstalling the Employee Delta Export Add-In for Microsoft Excel

Uninstall the Employee Delta Export Add-In for Microsoft Excel from your local computer if you no longer need it.

### Procedure

1. Go to Microsoft Excel and open the [SuccessFactors](#) ribbon tab.
2. Choose [About](#).
3. Choose [Uninstall](#) in the upcoming popup window.



## 4 Working with Employee Delta Export for Microsoft Excel

How to use the Employee Delta Export Add-In for Microsoft Excel.

[First Steps to Using the Employee Delta Export for Microsoft Excel \[page 17\]](#)

Take the first steps to start retrieving data using the Employee Delta Export.

[Architecture Concept of the Employee Delta Export for Microsoft Excel \[page 19\]](#)

More about the architecture of the Employee Delta Export Add-In for Microsoft Excel.

[Workbooks for Employee Delta Export for Microsoft Excel \[page 28\]](#)

The Employee Delta Export requires specific Microsoft Excel workbooks to fill in the data, which you can download from the *Employee Delta Export* UI.

[Worksheets of Employee Delta Export for Microsoft Excel \[page 29\]](#)

Employee Delta Export for Microsoft Excel comes with multiple worksheets, such as result, selection, or summary sheets.

[Tabs on the Ribbon \[page 48\]](#)

The Employee Delta Export Add-In for Microsoft Excel creates a *SuccessFactors* tab and adds a *SuccessFactors* group to the *Developer* tab in the Microsoft Excel ribbon.

### 4.1 First Steps to Using the Employee Delta Export for Microsoft Excel

Take the first steps to start retrieving data using the Employee Delta Export.

#### Procedure

1. Install the add-in.
2. Download the *Multi-Sheet Workbook* from the *Employee Delta Export* UI.
3. Open the *Reporting* workbook.
4. Go to the *Systems* worksheet and enter the system you want to retrieve data from:
  - *Name*: Free text to identify the system
  - *URL*: Enter the relevant API endpoint URL
  - *Company*: The name of the company (case sensitive)
5. Go to the *Selection* worksheet.
6. Choose the desired system.
7. Set the other input fields as follows:

New Report Start Date [UTC]	01/01/2014
New Report Start Time [UTC]	00:00:00

- Go to the [SuccessFactors](#) tab on the Excel ribbon and choose [Get Data](#). This calls the `CompoundEmployee` API in the SAP SuccessFactors system to return all changes that happened between 01.01.2014 and the current time.

A logon dialog pops up where you have to enter the following OAuth 2.0 keys:

- [User ID](#)
- [API Key](#)
- encrypted [Private Key](#) from the X.509 Certificate

This calls the `CompoundEmployee` API in the SAP SuccessFactors system to return all changes that happened between 01.01.2014 and the current time.

- Once the data is read, the view switches to the [Summary](#) sheet showing you the selection parameters and the number of records found.

#### [API Endpoint URLs for Employee Delta Export \[page 18\]](#)

Find the API endpoint to be used in the Employee Delta Export for Microsoft Excel. It depends on where your SAP SuccessFactors instance is located.

#### [Carrying Out an Example Test Run \[page 19\]](#)

Carry out a first test run of the Employee Delta Export for Microsoft Excel.

## Related Information

[Setting up the OAuth 2.0 Application \[page 11\]](#)

### 4.1.1 API Endpoint URLs for Employee Delta Export

Find the API endpoint to be used in the Employee Delta Export for Microsoft Excel. It depends on where your SAP SuccessFactors instance is located.

The API endpoint can be in one of several data centers. The endpoints by data center are listed below. For the Employee Delta Export Add-In for Microsoft Excel, use the base URL part from the SF-API endpoint URL. For example: `https://api2.successfactors.eu`

## 4.1.2 Carrying Out an Example Test Run

Carry out a first test run of the Employee Delta Export for Microsoft Excel.

### Procedure

1. Check the data sheets in the Employee Delta Export workbook for new content.
2. In the Employee Central system, change the address of an employee (for example, change the house number).
3. Go back to the Employee Delta Export workbook. On the [SuccessFactors](#) tab, select [Get Data](#).
4. Check the data sheet for changes and you will see exactly the one address change you added before.

## 4.2 Architecture Concept of the Employee Delta Export for Microsoft Excel

More about the architecture of the Employee Delta Export Add-In for Microsoft Excel.

### [Component Structure \[page 19\]](#)

Multiple components are involved in employee delta reporting using the Employee Delta Export Add-In for Microsoft Excel.

### [Data Extraction Process \[page 20\]](#)

Data is extracted by the Employee Delta Export Add-in for Microsoft Excel from Employee Central in three main steps.

### [How Delta Calculation Works \[page 20\]](#)

You can see the results the Employee Delta Export returns in more detail to set up business processes to post-process the retrieved content.

### 4.2.1 Component Structure

Multiple components are involved in employee delta reporting using the Employee Delta Export Add-In for Microsoft Excel.

The following components are involved in employee delta reporting:

- Microsoft Excel Application
- Microsoft Excel Add-in

The Microsoft Excel add-in is the heart of Employee Delta Export. It contains VBA code which performs the data extraction and delta calculation. The add-in must be installed into the Microsoft Excel application once and becomes an intrinsic component of Microsoft Excel. However, it remains inactive unless a reporting workbook is opened. The VBA code calls the `CompoundEmployee` API to receive the information about new and changed employees. In addition, it uses other direct and OData APIs to retrieve lookup data.

- Microsoft Excel reporting workbooks containing configuration information  
The reporting workbooks contain the result sheets that receive the data retrieved from the system plus configuration data controlling the extraction process.
- CompoundEmployee API  
The CompoundEmployee API is a SOAP API that extracts the data from the system. It is called by the VBA code of the add-in multiple times.
- Employee Central  
This product keeps all the employee data plus historical data about changes. It is accessed by the CompoundEmployee API.

## 4.2.2 Data Extraction Process

Data is extracted by the Employee Delta Export Add-in for Microsoft Excel from Employee Central in three main steps.

### Note

The Employee Delta Export is based on the CompoundEmployee API, which is a SOAP API. This means that the result returned by the API is in XML format. This is the reason why many configurations in the reporting workbooks are based on XPath syntax, since XPath is the perfect language to access elements in an XML document.

1. The Employee Delta Export application calls the CompoundEmployee API in delta mode to retrieve all creations, changes, and deletions of entities that happened since the start of the evaluation period that is typically the time of the previous extraction.  
Additional selection parameters going into the WHERE clause of the API call can be specified on the [Selection](#) worksheet. In case the first call in delta mode didn't retrieve all records matching the selection criteria, subsequent queryMore calls are issued to retrieve the remaining data.  
Depending on the version of the Employee Delta Export, subsequent calls to the CompoundEmployee API are performed. For the persons returned by the call in delta mode the API is called again in the so-called snapshot mode. This mode returns the status of all entities for these persons at the time of the previous extraction. After that, a third API call in full transmission mode is performed returning the current status of the persons in question.  
The raw data returned by all API calls is stored in the workbook on hidden sheets.
2. The client-side filters defined on the [Config](#) sheet are applied to the result XML returned from the system in order to exclude more employees from the result. Both filters are applied to the full data of each employee retrieved from the system including all time slices (filters aren't applied to the delta result).
3. The add-in post-processes the data and calculates the changes compared to the previous extraction on field level. Only changed fields are shown on the result sheets of the workbook unless it's enforced to show the current field values.

## 4.2.3 How Delta Calculation Works

You can see the results the Employee Delta Export returns in more detail to set up business processes to post-process the retrieved content.

The Employee Delta Export returns all changes that have been updated in the system or modified by APIs during the evaluation period of the report. The start of the evaluation period is as you specify in the fields [New Report Start](#)

*Date [UTC]* and *New Report Start Time [UTC]* on the *Selection* sheet. The end of the evaluation period is the point in time when the report is executed. The report returns all changes within this period no matter when they become effective.

First of all, you need to distinguish between data that is effective dated and data that isn't effective dated in Employee Central. Both types of data appear in the Microsoft Excel reporting workbooks. For example, personal information and job information are effective dated, whereas national ID card information and spot bonus information aren't.

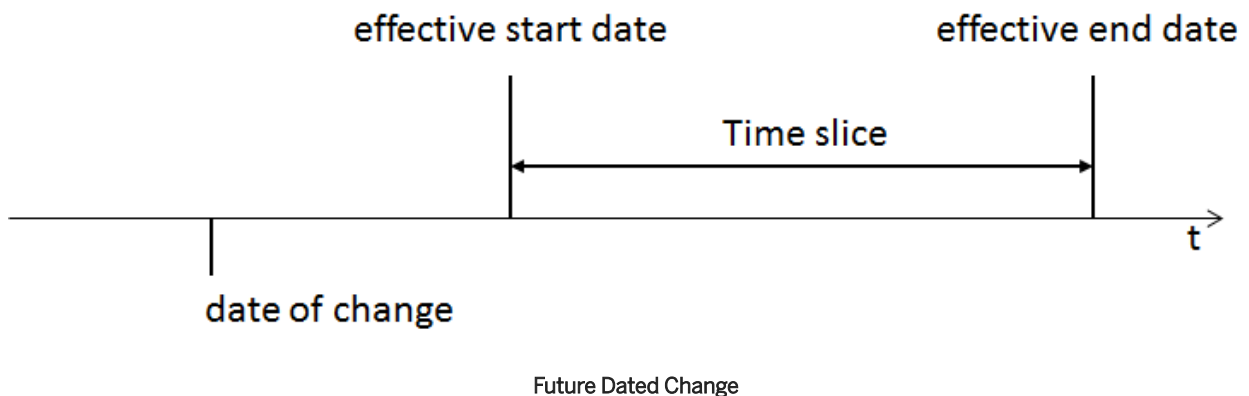
## Non-Effective-Dated Information

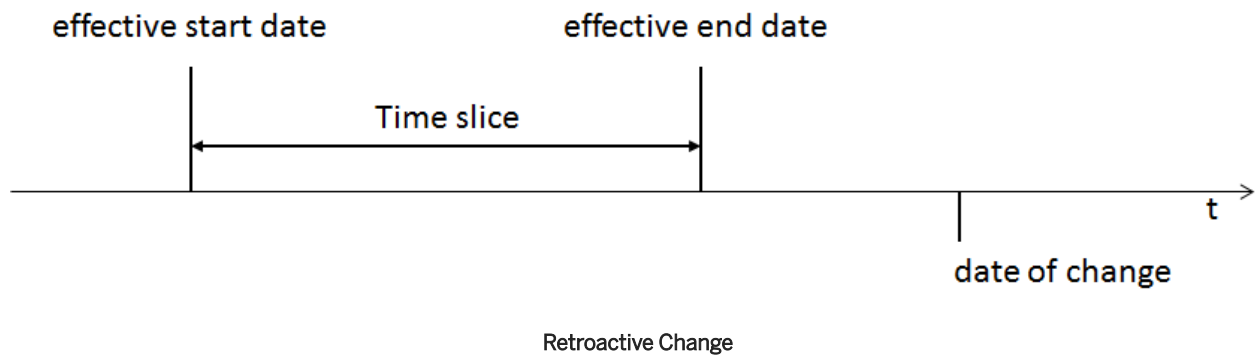
This type of information is easy to understand. Creations, changes, and deletions of such data can't be scheduled and become effective immediately for the whole time bar. Spot bonus (pay component non-recurring) is somewhat of an exception, because this is information that applies for one particular date only.

## Effective-Dated Information

For effective-dated information, you need to distinguish between the point in time when a change is made in the system and the point in time when this change becomes effective. Both points in time are independent of each other. A change made today can become effective today, in the future, or in the past. Changes that become effective in the past are called retroactive changes. If a change becomes effective in the past, it may no longer be effective today since the period of effectiveness could also end in the past. The time period during which a set of data is effective is called a time slice. The date when a change becomes effective is called the start date (the beginning of the time slice). The date when it ends being effective is called the end date (the end of the time slice).

The Employee Delta Export returns data that has been changed during the evaluation period of the report no matter when these changes become effective. This is important to know because in some cases you get information that you're not interested in at the point in time you get it. For example, if the promotion of an employee is made today, but becomes effective at the start of next year, you'll see this change in the next Microsoft Excel report run after the change was made. You need to take up this issue organizationally, for example, by shelving this change.





## Actions Performed in the System

From a business perspective, changes shown in the report can result from the following actions performed in the system:

- Hire of an employee
- Insertion of a new record for non effective-dated data
- Change of data of a non effective-dated entity
- Change of the key of an entity
- Insertion of a new time slice for effective-dated data
- Change of data in the history of effective-dated data using Make Correction
- Change of start date of effective-dated data in the history using Make Correction
- Deletion of a non effective-dated record (for example, personal information)
- Deletion of a record in the history
- Deletion of a record starting from a particular date (for example, home address)
- Deletion of a record in the course of changing the superordinate record

### [Delta Calculation for New Hire or Rehire \[page 23\]](#)

Here is more information about how delta calculation in the Employee Delta Export Add-In works when a new employee is hired.

### [Delta Calculation for Non-Effective-Dated Data \[page 23\]](#)

Here is more information about how delta calculation in the Employee Delta Export Add-In works for data that's not effective-dated.

### [Delta Calculation for Effective-Dated Data \[page 25\]](#)

Here is more information about how delta calculation in the Employee Delta Export Add-In works for data that's effective-dated.

### [Delta Calculation for Change of Entity Key \[page 28\]](#)

Here is more information about how delta calculation in the Employee Delta Export Add-In works if the key of an entity is changed.

## Related Information

[How Changes in the Result Sheets Are Highlighted \[page 33\]](#)

### 4.2.3.1 Delta Calculation for New Hire or Rehire

Here is more information about how delta calculation in the Employee Delta Export Add-In works when a new employee is hired.

A hire can be a new hire or a rehire. Both are shown in the same way so that the complete data of all entities and all time slices are shown in the report. In this case, the status columns have the value *New*. The *[Old]* columns stay empty.

	A	B	L	M	X	Y
4	Personal Information					
5	Status	User Id	Last Name [New]	Last Name [Old]	Marital Status [New]	Marital Status [Old]
6	New	20010	Kendrick		S	
7	New	20030	Scott		D	
8	New		Scott		M	
9	New	20040	Anderson		D	
10	New	20050	Walker		S	

Example: Newly Hired or Rehired Employee

#### Note

The highlighting shown in the example can differ from the highlighting in the version of the add-in you've installed.

### 4.2.3.2 Delta Calculation for Non-Effective-Dated Data

Here is more information about how delta calculation in the Employee Delta Export Add-In works for data that's not effective-dated.

#### Inserting New Record

New records of non-effective-dated data (for example, national ID card) are shown as new rows in the Microsoft Excel report with value *NewSeg* in the status column. The *[Old]* columns stay empty.

	A	B	D	E	F	G	H	I	J
4	National ID Information								
5	Status	User Id	First Name	Last Name	Country	National ID Type [New]	National ID Type [Old]	National ID [New]	National ID [Old]
6	Changed	20030	John	Scott	USA	ssn		235-20-0030	234-20-0030
7	NewSeg	20050	Patricia	Walker	CAN	sin		123-654-234	
8	New	20120	Robert	Lopez	USA	ssn		234-20-0120	
9	New	20130	Michael	Lee	USA	ssn		234-20-0130	
10	New	20140	Elizabeth	Mitchell	USA	ssn		234-20-0140	

Example: New Record of Non-Effective-Dated Data

## Changing Non-Effective-Dated Data

A simple change of non-effective-dated data is shown as new rows in the Microsoft Excel report with the value *Changed* in the status column. The *[New]* columns show the latest values whereas the *[Old]* columns show the values valid before the change happened.

## Deleting Record

When you delete a non-effective-dated record in the system, this is shown in the Microsoft Excel report as new rows with the value *Deleted* in the status column. The *[Old]* columns don't change. The *[New]* columns contain the pseudo value *deleted*. Row 8 in the following example shows such a case.

	A	B	D	E	F	G	H	I	J
4	National ID Information								
5	Status	User Id	First Name	Last Name	Country	National ID Type [New]	National ID Type [Old]	National ID [New]	National ID [Old]
6	Changed	20030	John	Scott	USA	ssn		235-20-0030	234-20-0030
7	NewSeg	20050	Patricia	Walker	CAN	sin		123-654-234	
8	Deleted				USA	ssn		***deleted**	234-20-0320
9	New	20120	Robert	Lopez	USA	ssn		234-20-0120	

Example: Deletion of Non-Effective-Dated Record

### Note

The highlighting shown in the examples can differ from the highlighting in the version of the add-in you've installed.



### 4.2.3.3 Delta Calculation for Effective-Dated Data

Here is more information about how delta calculation in the Employee Delta Export Add-In works for data that's effective-dated.

#### Inserting New Time Slice

If you insert a new time slice in the system using the *Insert Record* function, this is shown in the Microsoft Excel report as a separate row with the value *Changed* in the status column. The *[Old]* columns show the values that were valid for the validity period before the new time slice was inserted.

#### Changing Data in the History Using Make Correction

This kind of change is shown in the Microsoft Excel report just like the insertion of a new time slice. The *[Old]* columns show the values that were valid before the correction was performed. Row 6 of the example shows such a case.

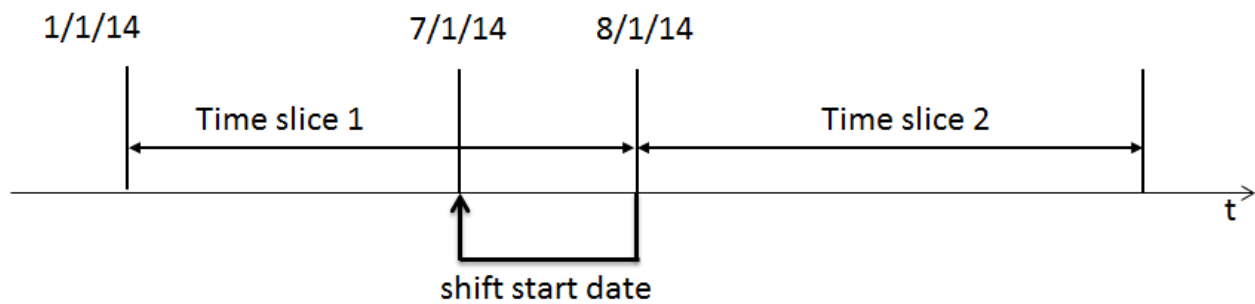
	A	B	D	E	F	G	J	K	L	M	
4	Employee Address										
5	Status	User Id	First Name	Last Name	Start Date	End Date	Address Type	Country	Address1 [New]	Address1 [Old]	Ado [Ne
6	Changed	20030	John	Scott	01.09.2014	31.12.9999	Home		66 Greenway Park Rd	1084 Foster City Blvd	
7	NewSeg	20040	Kevin	Anderson	01.09.2014	31.12.9999	Home	USA	66 Greenway Park Rd		
8	Changed	20050	Patricia	Walker	11.07.2014	17.07.2014	Home		3541 Wheeler Rd	3540 Wheeler Rd	
9	Changed				18.07.2014	31.12.9999	Home		211 Chambers St	3540 Wheeler Rd	
10	Changed	20060	Kimberly	Smith	18.07.2014	17.08.2014	Home				
11	Changed				18.08.2014	31.12.9999	Home		664 Scranton Rd	1111 US Highway 441 N	
12	Deleted	20070	Donna	Evans	11.07.2014	30.12.9999	Home	USA	deleted	117 Kite Rd	
13	New	20120	Robert	Lopez	11.07.2014	31.12.9999	Home	USA	1234 Creekside Place		

Example: Change Using Make Correction

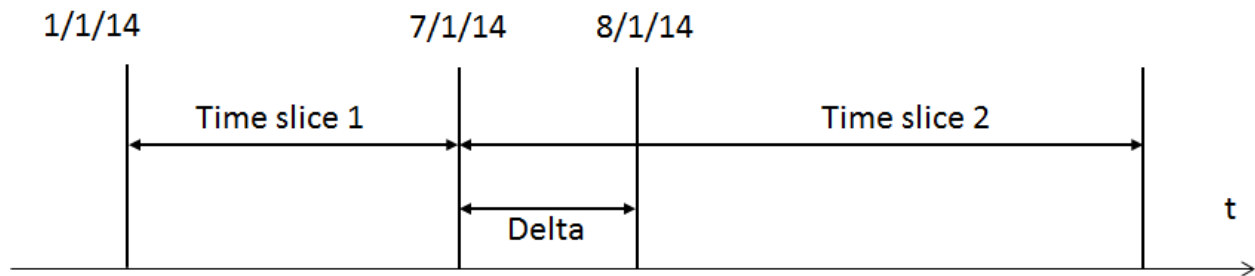
#### Changing Start Date in the Past Using Make Correction

This is the most complex scenario. If the start date of a time slice is shifted to the future or the past, this is shown in the Microsoft Excel report as a separate row. The row shows the differences between the old and the new values for the period that is affected by the shift.

In the graphics, the start date of time slice 2 is changed from January 8, 2014 to January 7, 2014. As a result, the Microsoft Excel report shows a new row with the start date January 7, 2014 and the end date July 31, 2014 (denoted as *Delta* in the figure). This row shows the values of time slice 2 in the *[New]* columns and the values of time slice 1 in the *[Old]* columns.



Start Date Changed in the Past

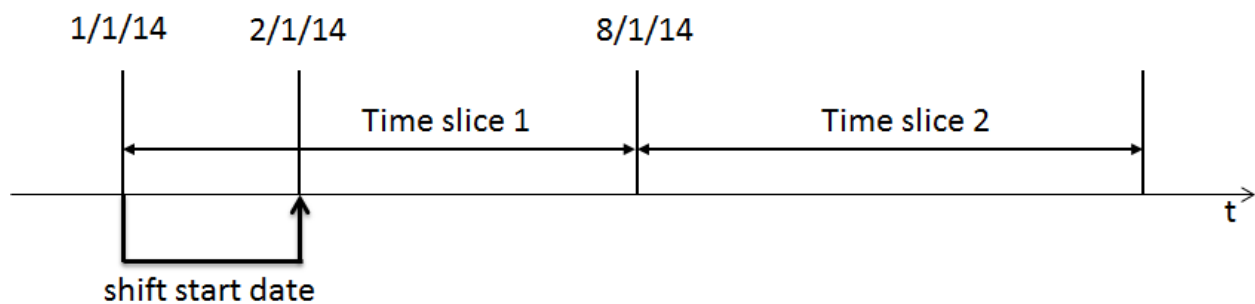


Delta When Start Date Changed in the Past

In cases where the start date of the first time slice is moved to the future, the Microsoft Excel report shows a row denoting a deletion of the delta period.

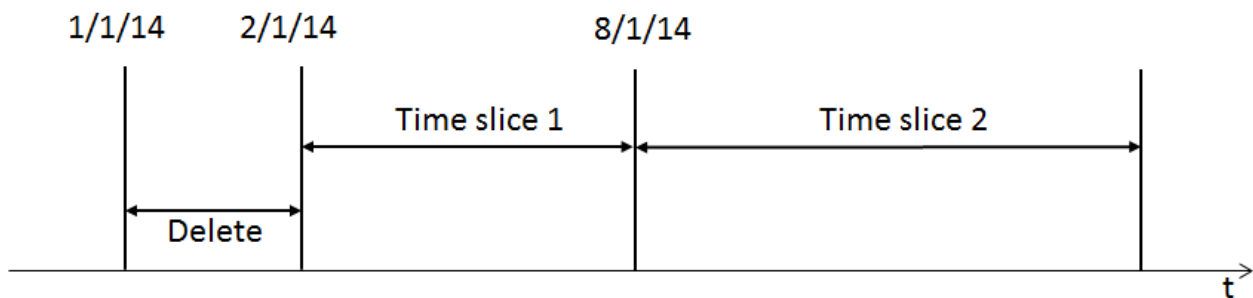
	A	B	D	E	F	G	L	M	N
4	Employee Address								
5	Status	User Id	First Name	Last Name	Effective As Of (Start Date)	End Date	Country [New]	Country [Old]	Address1 [New]
6	Deleted	20070	Donna	Evans	11.07.2014	30.12.9999	***deleted***	USA	***deleted***

Example: Deletion of Delta Period



Start of First Time Slice Moved to the Future

The result of the shift is shown here:



Result of Moving Start of First Time Slice to the Future

## Deleting Record in the History

When you delete an effective-dated record in the system, this is shown in the Microsoft Excel report as new rows with the value *Deleted* in the status column. The *[Old]* columns don't change. The *[New]* columns contain the pseudo value *deleted*.

## Deleting an Address as of a Specific Date

Assume that you choose *Edit* in the *Address* entity and delete, for example, the vacation address after entering the effective start date. In the Microsoft Excel report, this is shown as a new address row with value *Deleted* in the status column. The *[Old]* columns show the values just before the deletion. The *[New]* columns contain the pseudo value *deleted*.

## Deleting Record in the Course of Changing the Superordinate Data

A typical example is that you create a new compensation time slice with the start date X. In the course of that, you delete a pay component from the list. In the Microsoft Excel report, this is shown as a pay component row with value *Deleted* in the status column. The *[Old]* columns show the values just before the deletion. The *[New]* columns contain the pseudo value *deleted*. Row 9 in the example shows such a deleted pay component.

	A	B	C	D	F	G	H	K	L	M	N	O	P	
4	Pay Component Recurring Information													
5	Status	User Id	First Name	Last Name	Pay Component	Start Date	End Date	Currency [New]	Currency [Old]	Frequency [New]	Frequency [Old]	Amount [New]	Amount [Old]	Last
6	Changed	20220	Thomas	Baker	Base Salary	01.09.2014	31.12.2014	USD	USD	MON	MON	5000,00	5500,00	01.0
7	NewSeg				Monthly Car	01.09.2014	31.12.2014	USD	USD	MON		200,00		01.0
8	Changed				Base Salary	01.01.2015	31.12.9999	USD	USD	MON	MON	5000,00	5500,00	30.0
9	Deleted	20130	Michael	Lee	Monthly Car	01.11.2014	31.12.9999	***deleted	USD	***deleted*	MON	***deleted*	200,00	
10	New	20140	Elizabeth	Mitchell	Bonus	11.07.2014	31.12.9999	USD		MON		20,00		01.0

Example: Record Deleted by Change to Superordinate Data

### Note

The highlighting shown in the examples can differ from the highlighting in the version of the add-in you've installed.

## 4.2.3.4 Delta Calculation for Change of Entity Key

Here is more information about how delta calculation in the Employee Delta Export Add-In works if the key of an entity is changed.

If a key field of a non-effective-dated entity or effective-dated entity is changed in the system (for example, the country/region of a national ID card), the old record with the old key is deleted and a new record with the new key is created. The Microsoft Excel report shows a row with status *Deleted* for the first action and a row with status *NewSeg* for the second action. Rows 7 and 8 of the example show such a case.

	A	B	D	E	F	G	H	I	J
4	National ID Information								
5	Status	User Id	First Name	Last Name	Country	National ID Type [New]	National ID Type [Old]	National ID [New]	National ID [Old]
6	Changed	20030	John	Scott	USA	ssn		235-20-0030	234-20-0030
7	NewSeg	20050	Patricia	Walker	CAN	sin		123-654-234	
8	Deleted				USA	ssn		***deleted***	234-20-0320
9	New	20120	Robert	Lopez	USA	ssn		234-20-0120	

Example: Key Change

### Note

The highlighting shown in the example can differ from the highlighting in the version of the add-in you've installed.

## 4.3 Workbooks for Employee Delta Export for Microsoft Excel

The Employee Delta Export requires specific Microsoft Excel workbooks to fill in the data, which you can download from the [Employee Delta Export](#) UI.

An Employee Delta Export workbook has a specific structure and contains configuration information that controls the behavior of the add-in. We provide two different workbook templates you can use as a starting point for customization:

- **CompEmpDelta\_multiSheet.xlsx**  
Contains one sheet for each entity that is supported by the application. Each sheet shows new and changed data in combination.
- **CompEmpDelta\_singleSheet.xlsx**  
Contains three data sheets:
  - New employees (hired, rehired, or transferred employees)
  - Changed employee data
  - Leaving employees

## 4.4 Worksheets of Employee Delta Export for Microsoft Excel

Employee Delta Export for Microsoft Excel comes with multiple worksheets, such as result, selection, or summary sheets.

Here's an example of what a worksheet might look like:

	A	B	D	E	F	G	J	K	L	M	N
4	Personal Information										
5	Status	User ID	First Name	Last Name	Effective As Of (Start Date)	End Date	First Name (New)	First Name (Old)	Last Name (New)	Last Name (Old)	Middle Name (New)
6	New	20010	Amy	Kendrick	11.07.2014	11.12.9999	Amy		Kendrick		
7	New	20030	John	Scott	11.07.2014	24.08.2014	John		Scott		
8	New	*			25.08.2014	11.12.9999	John		Scott		
9	New	20040	Kevin	Anderson	11.07.2014	11.12.9999	Kevin		Anderson		
10	New	20050	Patricia	Walker	11.07.2014	11.12.9999	Patricia		Walker		
11	New	20060	Kimberly	Smith	11.07.2014	11.12.9999	Kimberly		Smith		
12	New	20070	Donna	Evans	11.07.2014	11.12.9999	Donna		Evans		
13	New	20080	Mark	Turner	11.07.2014	11.12.9999	Mark		Turner		
14	New	20090	Ethan	White	11.07.2014	11.12.9999	Ethan		White		
15	New	20100	James	Green	11.07.2014	11.12.9999	James		Green		
16	New	20110	Barbara	Williams	11.07.2014	11.12.9999	Barbara		Williams		
17	New	20120	Joseph	Wright	11.07.2014	30.09.2014	Joseph		Wright		
18	New	*			01.10.2014	11.12.9999	Joseph		Wright		Alexander
19	New	20200	Laura	Edwards	11.07.2014	11.12.9999	Laura		Edwards		
20	New	20210	Sandra	Nelson	11.07.2014	11.12.9999	Sandra		Nelson		
21	New	20220	Thomas	Baker	11.07.2014	11.12.9999	Thomas		Baker		
22	New	20240	Justin	Lewis	11.07.2014	11.12.9999	Justin		Lewis		
23	New	20250	Todd	Allen	11.07.2014	11.12.9999	Todd		Allen		
24	New	20260	Jesse	Young	11.07.2014	11.12.9999	Jesse		Young		
25	New	20270	Sean	Hernandez	11.07.2014	11.12.9999	Sean		Hernandez		
26	New	20280	Earl	Hall	11.07.2014	11.12.9999	Earl		Hall		
27	New	20130	Robert	Lopez	11.07.2014	11.12.9999	Robert		Lopez		
28	New	20140	Michael	Lee	11.07.2014	11.12.9999	Michael		Lee		
29	New	20140	Elizabeth	Mitchell	11.07.2014	11.12.9999	Elizabeth		Mitchell		

Personal Info Sheet with Changes and New Data of Multiple Employees

### Result Sheets [\[page 30\]](#)

The result sheets of Employee Delta Export for Microsoft Excel are populated by the data retrieved from the backend by the `Compound Employee API`.

### Selection Sheet [\[page 34\]](#)

Employee Delta Export for Microsoft Excel has a [Selection](#) sheet, which contains parameters used to filter data.

### Summary Sheets [\[page 42\]](#)

The [Summary](#) sheets of Employee Delta Export for Microsoft Excel give a summary of the last extraction run.

### Systems Sheet [\[page 42\]](#)

The Employee Delta Export for Microsoft Excel has a [Systems](#) sheet, which contains all the systems to which you can connect.

### Config Sheet [\[page 43\]](#)

The [Config](#) sheet of the Employee Delta Export for Microsoft Excel contains configuration parameters that control how the extraction is performed and how the results are displayed.

### Errors Sheet [\[page 46\]](#)

Employee Delta Export for Microsoft Excel has an [Errors](#) sheet, which shows all employees who could not be replicated for whatever reason.

### Warnings Sheet [\[page 47\]](#)

The [Warnings](#) sheet of the Employee Delta Export for Microsoft Excel shows employees for which certain fields could not be retrieved due to picklist or foundation object configuration issues.

### Internal Sheets [\[page 47\]](#)

A couple of worksheets in the reporting workbook of the Employee Delta Export for Microsoft Excel are used for internal purposes and are not shown to the end user unless you make them visible.

## 4.4.1 Result Sheets

The result sheets of Employee Delta Export for Microsoft Excel are populated by the data retrieved from the backend by the `CompoundEmployee` API.

Employee Delta Export for Microsoft Excel provides two sample reporting workbooks, [CompEmpDelta\\_singleSheet.xlsx](#) and [CompEmpDelta\\_multiSheet.xlsx](#), which have different sets of result sheets. In both cases the result sheets have a common layout in the first 4 columns:

Column 1: Status reflecting the reason why the row appears in the workbook

Column 2: User ID of the employee

Column 3: First name of the employee no matter if it was changed

Column 4: Last name of the employee no matter if it was changed

The result sheets have two columns for data fields that may change. The column with suffix [\[New\]](#) contains the current value and the columns with suffix [\[Old\]](#) contains the value the field had at the start of the evaluation period. Note, however, that if the data field belongs to an effective-dated entity the [\[Old\]](#) value is subject to the start and end date in the same row.

### [Result Sheets of the CompEmpDelta\\_singleSheet.xlsx Sample Workbook \[page 30\]](#)

This `CompEmpDelta_singleSheet.xlsx` workbook has different result sheets for hires, changes, and leavers. On each of the result sheets you find the data of all Employee Central entities supported by the Employee Delta Export.

### [Result Sheets of the CompEmpDelta\\_multiSheet.xlsx Workbook \[page 31\]](#)

The `CompEmpDelta_multiSheet.xlsx` workbook contains separate result worksheets for all Employee Central entities supported by the Employee Delta Export.

### [How Changes in the Result Sheets Are Highlighted \[page 33\]](#)

The Employee Delta Export Add-In for Microsoft Excel uses highlighting in the result sheet to make it easier for you to see what has changed.

### 4.4.1.1 Result Sheets of the CompEmpDelta\_singleSheet.xlsx Sample Workbook

This `CompEmpDelta_singleSheet.xlsx` workbook has different result sheets for hires, changes, and leavers. On each of the result sheets you find the data of all Employee Central entities supported by the Employee Delta Export.

Result Sheets for Hires, Changes, And Leavers

Sheet	Description
Hires	The <a href="#">Hires</a> sheet always shows the complete set of data for the hired or rehired employee. All time slices are shown, including time slices in the future.

Sheet	Description
Changes	<p>The <a href="#">Changes</a> sheet shows employees whose data have been changed in the system during the evaluation period. Only time slices that have been changed are shown in this case.</p> <p>Depending on the configuration parameter <a href="#">Show Unchanged Values</a>, the sheet shows all values of a changed entity or only those that have actually been modified. The status column distinguishes between the different events causing the change. It contains one of the following values:</p> <ul style="list-style-type: none"> <li>Changed</li> </ul> <p>Indicates a change during the evaluation period. This can be caused by either a correction or by adding a new time slice.</p> <ul style="list-style-type: none"> <li>NewSeg</li> </ul> <p>Indicates that the entity record was created during the evaluation period. For example, a new national ID card entry was added or a new pay compensation entry was added. NewSeg is not used, however, if a new time slice was added to an existing entry, like personal information. This case is portrayed as a change instead.</p> <ul style="list-style-type: none"> <li>Deleted</li> </ul> <p>Indicates that the time slice or an entity record was deleted during the evaluation period.</p>
Leavers	<p>The <a href="#">Leavers</a> sheet shows only those employees for whom a termination event has been added during the evaluation period. Only limited data is shown for these employees.</p>

Each worksheet contains the data of all Employee Central entities related to the employee. Each entity is shown in a separate row. For effective-dated entities, each time slice is shown in a separate row. Rows belonging to the same employee are grouped by color. The column [Start Date](#) in the column section for a particular effective-dated entity shows the start date of the time slice.

#### 4.4.1.2 Result Sheets of the CompEmpDelta\_multiSheet.xlsx Workbook

The CompEmpDelta\_multiSheet.xlsx workbook contains separate result worksheets for all Employee Central entities supported by the Employee Delta Export.

The Employee Delta Export for Microsoft Excel supports the following entities:

- Personal Information
- National ID Information
- Address Information

- Employment Details
- Job Information
- Pay Compensation Recurring
- Pay Compensation Non-Recurring (Spot Bonus)
- Payment Information
- Deduction Recurring
- One Time Deduction
- Alternative Cost Distribution
- Personal Documents Information

Each sheet contains data for hires, rehires, changes, and leaves. For effective-dated entities each time slice is shown in a separate row. Rows belonging to the same employee are grouped by color.

The different events are distinguished by the status column. The status column contains one of the values below:

#### Status Values for Events

<b>New</b>	<b>A hire or rehire has been entered during the evaluation period. The full set of data including all time slices is returned.</b>
<b>Changed</b>	This can be caused by either a correction or by adding a new time slice.
<b>NewSeg</b>	A new entity record was created during the evaluation period. For example, a new national ID card entry was added or a new pay compensation entry was added. NewSeg is not used, however, if a new time slice was added to an existing entry like for example personal information. This case is denoted as a change instead.
<b>Deleted</b>	The time slice was deleted during the evaluation period
<b>Transfer</b>	The employee has entered the population specified by the selection criteria during the evaluation period. This means that the employee didn't fulfill the selection criteria at the start of the evaluation period. In this case, the full set of data including all time slices is.

	A	B	D	E	F	G	J	L	M	N	O	
4	Employee Address											
5	Status	User Id	First Name	Last Name	Start Date	End Date	Address Type	Country [New]	Country [Old]	Address1 [New]	Address1 [Old]	Ad [Ne
6	Changed	20030	John	Scott	01.09.2014	31.12.9999	Home			66 Greenway Park Rd	1084 Foster City Blvd	
7	NewSeg	20040	Kevin	Anderson	01.09.2014	31.12.9999	Home	USA		66 Greenway Park Rd		
8	Changed	20050	Patricia	Walker	11.07.2014	17.07.2014	Home			3541 Wheeler Rd	3540 Wheeler Rd	
9	Changed				18.07.2014	31.12.9999	Home			211 Chambers St	3540 Wheeler Rd	
10	Changed	20060	Kimberly	Smith	18.07.2014	17.08.2014	Home					
11	Changed				18.08.2014	31.12.9999	Home			664 Scranton Rd	1111 US Highway 441 N	
12	Deleted	20070	Donna	Evans	11.07.2014	30.12.9999	Home	deleted	USA	deleted	117 Kite Rd	
13	New	20120	Robert	Lopez	11.07.2014	31.12.9999	Home	USA		1234 Creekside Place		

Example: Address Information Worksheet



### 4.4.1.3 How Changes in the Result Sheets Are Highlighted

The Employee Delta Export Add-In for Microsoft Excel uses highlighting in the result sheet to make it easier for you to see what has changed.

If changes on the result sheets populated by Employee Delta Export are on the field level, one or more single cells are highlighted. If changes affect the whole record on a result sheet, the status cell is highlighted.

In this example, the whole national ID record was deleted, which you can see here because the status cell is highlighted. In order to identify the deleted record, the business key stays unchanged and fields that don't belong to the business key are shown as **\*\*\*deleted\*\*\***.

	A	B	C	D	E	F	G	H	I
4	<b>National ID Information</b>								
5	Status	User Id	First Name	Last Name	Country	National ID Type	National ID [New]	National ID [Old]	Is Primary [New]
6	Deleted	person_id_external	first_name	last_name	country	card_type	***deleted***	national_id	***deleted***

Example: Whole Record Deleted

The following example shows another deletion, however, this time, only one value of the record was deleted. The changed value is highlighted.

	A	B	C	D	E	F	G	J	K
4	<b>National ID Information</b>								
5	Status	User Id	First Name	Last Name	Country	National ID Type [New]	National ID Type [Old]	Is Primary [New]	Is Primary [Old]
6	Changed	test01	first_name	last_name	USA	card_type		TRUE	FALSE

Example: One Value of Record Deleted

## 4.4.2 Selection Sheet

Employee Delta Export for Microsoft Excel has a [Selection](#) sheet, which contains parameters used to filter data.

Here's an example of what the [Selection](#) sheet might look like:

	A	B	C
1		<b>Employee Delta Export</b>	
2		Workbook Version: 1.18	Workbook Release: 1411
3		Add-in Version: 1411.0.0	Add-in Release: 1411
4		This workbook needs to be customized before using it productively	
5			
6		Choose System:	localhost
7			
8		New Report Start Date [UTC]	13.01.2015
9		New Report Start Time [UTC]	14:01:08
10			
11		Country	USA
12		Company (Legal Entity)	
13		Division	
14		Business Unit	
15		Department	
16		Cost Center	ABC
17		Work Location	
18		Pay Group	
19		Employee Class	
20		User Id	
21		Exclude User Ids	
22			
23		Last Report Start Date [UTC]	01.08.2014
24		Last Report Start Time [UTC]	
25		Last Report End Date [UTC]	13.01.2015
26		Last Report End Time [UTC]	14:01:08

Selection Sheet with Selection Parameters

### [Filter Parameters on the Selection Sheet \[page 35\]](#)

The [Selection](#) sheet of Employee Delta Export for Microsoft Excel contains parameters you can use to filter data.

### [How Filtering Works \[page 36\]](#)

Here's an overview of how the filter parameters work that are provided on the [Selection](#) sheet of the Employee Delta Export for Microsoft Excel.

### [How Period-Based Delta Works \[page 40\]](#)

You can use period-based delta on the [Selection](#) and [Summary](#) sheets to restrict reporting to those changes to data that are relevant for a given evaluation period, such as the time period to be considered in a payroll run.

### [How Effective Period Selection Works \[page 41\]](#)

You can use filter parameters on the [Selection](#) sheet to define effective period selection, thus selecting employees that fulfill certain conditions in a specific time frame.

## Related Information

### [How Period-Based Delta Works \[page 40\]](#)

## 4.4.2.1 Filter Parameters on the Selection Sheet

The [Selection](#) sheet of Employee Delta Export for Microsoft Excel contains parameters you can use to filter data.

### Parameters on the Selection Sheet

Choose System	The system to which the http request goes. The dropdown list shows the entries found on the Systems sheet.
New Report Start Date/Time [UTC]	<p>The start date/time of the evaluation period. All creations, changes, and deletions of entities applied in the backend since this point in time will go into the result sheets. Once the data has been retrieved from the backend completely, these two fields are automatically updated with the server time when you started the recent extraction. Unless you change these, this date/time is used for the next extraction run.</p> <p>When no changes are detected in the extraction run, the message <a href="#">No Changes Found</a> is issued and the fields remain unchanged.</p>
Last Report Start Date/Time [UTC]	<p>This is the start date/time of the previous extraction run. It is automatically updated after the extraction is performed.</p> <p>When no changes are detected in the extraction run, the message <a href="#">No Changes Found</a> is issued and the fields remain unchanged.</p> <p>This field is read-only.</p>
Last Report End Date/Time [UTC]	<p>This is the end date/time of the previous extraction. It is in fact the date/time when you pressed the <a href="#">Get Data</a> button the last time (in UTC time zone). It is typically identical with the <a href="#">New Report Start Date/Time</a> mentioned above.</p> <p>This field is read-only.</p>
Period Start Date	Marks the start of the validity period for which data is considered.
Period End Date	Marks the end of the validity period for which data is considered.
First Run in Period	Indicates whether this is the first report run in that respective period.

## Filter Parameters

The remaining parameters are used for filtering the employees when retrieving them from the backend or for post-processing. You can either specify a single value or a comma-separated list. For details please see the section [Filter Logic](#) .

Example:

The following parameter specification

Division	ABC
----------	-----

results in a where clause WHERE division = 'ABC' being sent to the API in the backend

The following parameter specification

Division	ABC, BCD
----------	----------

results in a where clause WHERE division IN ('ABC','BCD') being sent to the API in the backend

The following parameter specification

Division	ABC
Department	BCD

results in a where clause WHERE division = 'ABC' AND Department = 'BCD'.

The following parameter specification

Cost Center	ABC
-------------	-----

results in filtering for cost center ABC in job information during post-processing.

## 4.4.2.2 How Filtering Works

Here's an overview of how the filter parameters work that are provided on the [Selection](#) sheet of the Employee Delta Export for Microsoft Excel.

The [Selection](#) worksheet has two different kinds of filter parameters:

- Filter parameters that are passed through to the `CompoundEmployee` API call.
- Filter parameters that are applied during post-processing in Microsoft Excel locally.

To understand the result of the report, you need to understand how the filtering works for both cases.

## Filter parameters passed through to the Compound Employee API

There is a limited set of filter parameters provided by the CompoundEmployee API. These are:

- BUSINESS\_UNIT
- COMPANY\_TERRITORY\_CODE
- COMPANY
- DEPARTMENT
- DIVISION
- EMPLOYEE\_CLASS
- LOCATION
- JOB\_CODE
- PAY\_GROUP
- PERSON\_ID
- PERSON\_ID\_EXTERNAL
- USER\_ID

For these parameters you can either specify a single value or a comma-separated list of values on the [Selection](#) sheet. A single values results in the WHERE <field> = '<value>' condition passed to the API. A list results in a WHERE <field> IN ('value1', 'value2', ...) to be passed to the API.

If you specify values for multiple parameters on the [Selection](#) sheet, the filter conditions are logically ANDed in the API call.

### Note

Selection in the CompoundEmployee API works as follows: An employee counts as selected if each sub-condition is met by at least one of the segments. This may be unexpected since the sub-conditions don't need to be matched in the same segment.

### Example

You specify the following filter values:

- Department: abc
- Division: xyz

Both fields are contained in the Job Information segment. These conditions are met by an employee who has a single job info time slice that matches both conditions but also by an employee who has two job time slices where one meets the department condition and the other meets the division condition.

Filter parameters belonging to this category have an Microsoft Excel name along this pattern [EC\\_Filter\\_<parameter name>](#), for example: [EC\\_Filter\\_Company](#).

## Filter parameters applied during post-processing

Filter parameters belonging to this category have an Microsoft Excel name matching the pattern [EC\\_FilterAtClient\\_<parameter name>](#), for example, [EC\\_FilterAtClient\\_CostCenter](#).

To make these kinds of parameters effective, you need to include the filters into the sections [Additional Filters \(ANDed\)](#) and [Additional Filters \(ORed\)](#) on the [Config](#) sheet. The logic for the section [Additional Filters \(ANDed\)](#) is the same as for the system-related filter parameters. This means, filter expressions specified here and logically ANDed with the system-related filters and they apply if any of the segments meets the filter condition. As an example, the cost center filter is considered in the [Additional Filters \(ANDed\)](#) section.

Filter expressions in section [Additional Filters \(ORed\)](#) are applied to the employee who passed the previous checks. An employee meets the set of conditions in this section if any of the conditions are met.

The filter expressions in both [Additional Filter](#) sections may be complex XPath expressions containing multiple filter parameters. This allows for filters that check complex conditions on the same segment and so overcome the restriction mentioned above.

### Note

You can use a comma-separated value list only with one parameter of a complex condition due to complexity reasons. An error will be raised if you ignore this.

## Example 1

You specify the following filter values:

- Department: abc
- Cost Center: xyz

This filters employees who have department abc in one job segment and cost center xyz in the same or a different job segment.

## Example 2

You specify the following filter values:

- Department: abc
- Cost Center: xyz, uvw

This filters employees who have department abc in one job segment and cost center xyz or uvw in the same or a different job segment.

## Example 3

You specify the following filter values:

- Department: abc,def
- Cost Center: xyz, uvw

This filters employees who have department abc or def in one job segment and cost center xyz or uvw in the same or a different job segment.

## Example 4

You specify the following filter values, where Region is a [custom\\_string1](#) of job info and the Microsoft Excel name for the filter parameter in the [Selection](#) sheet is EC\_FilterAtClient\_Region.

- Region: abc
- Cost Center: xyz, uvw
- Additional filter (ANDed)://job\_information[cost\_center='%EC\_FilterAtClient\_CostCenter%']

- Additional filter (ANDed): //job\_information[custom\_string1='%EC\_FilterAtClient\_Region%']

This filters employees who have region abc in one job segment and cost center xyz or uvw in the same or a different job segment.

### Example 5

You specify the following filter values where Region is a *custom\_string1* of job information and the Microsoft Excel name for the filter parameter in the *Selection* sheet is EC\_FilterAtClient\_Region.

- Region: abc
- Cost Center: xyz, uvw
- Additional filter (ANDed): //job\_information[cost\_center='%EC\_FilterAtClient\_CostCenter%' AND custom\_string1='%EC\_FilterAtClient\_Region%']

This filters employees who have region abc in one job segment and cost center xyz or uvw in the job segment.

### Example 6

You specify the following filter values, where Region is a *custom\_string1* of job information and the Microsoft Excel name for the filter parameter in the *Selection* sheet is EC\_FilterAtClient\_Region.

- Region: abc,def
- Cost Center: xyz,uvw
- Additional filter (ANDed): //job\_information[cost\_center='%EC\_FilterAtClient\_CostCenter%' AND custom\_string1='%EC\_FilterAtClient\_Region%']

This combination of filter and filter values is invalid since both parameters in the filter expression are multi-valued.

### Example 7

You specify the following filter values, where Region is a *custom\_string1* of job info and the Microsoft Excel name for the filter parameter in the *Selection* sheet is EC\_FilterAtClient\_Region.

- Region: abc
- Cost Center: xyz
- Additional filter (ORed): //job\_information[cost\_center='%EC\_FilterAtClient\_CostCenter%']
- Additional filter (ORed): //job\_information[custom\_string1='%EC\_FilterAtClient\_Region%']

This filters employees who have a job info segment with region abc or a job segment with cost center xyz or both.

## Related Information

[Adding New Filter Parameters \[page 67\]](#)

[Employee Central Compound Employee API](#)

### 4.4.2.3 How Period-Based Delta Works

You can use period-based delta on the [Selection](#) and [Summary](#) sheets to restrict reporting to those changes to data that are relevant for a given evaluation period, such as the time period to be considered in a payroll run.

Period-based delta within the Employee Delta Export for Microsoft Excel considers the following kinds of changes:

- Changes to effective-dated entities that are valid in the past, or whose validity overlaps with the given period and happened after last synchronization.
- Changes on /creation of effective-dated entities in the past and where the start date falls into the given period.
- Changes on non-effective-dated entities happened after last synchronization.

The following fields show period-based delta information on the [Selection](#) and [Summary](#) sheets:

- [Period Start Date](#)  
Marks the start of the validity period for which data is considered.
- [Period End Date](#)  
Marks the end of the validity period for which data is considered.
- [First Run in Period](#)  
Indicates that this is the first report run in the respective period.

#### Note

Leave all three fields empty to run the Employee Delta Export without period-based delta.

### Use Case: Preparing upcoming payroll runs

A typical use case for period-based delta is when you are using the Employee Delta Export to prepare employee data for an upcoming payroll run, in which case you would use [Period Start Date](#) and [Period End Date](#) to mark the relevant start and end dates of the payroll run in question.

The payroll run is scheduled for June 25 and the data to be considered for this run is all data effective between June 1 and June 30.

On June 20, take a snap shot of the data in the Employee Central system that you want to update in the payroll system. In this example, the last snapshot was taken on May 20 at 9:30:40 UTC. Enter the following data on the [Selection](#) sheet:

New Report Start Date [UTC]	20.05.2015
New Report Start Time [UTC]	09:30:40
Period Start Date	01.06.2015
Period End Date	30.06.2015
First Run in Period	Yes

You execute this run on June 20, 2015 at 09:40:50 UTC and you want to perform a run for a second update of data in the payroll system for the same period. Enter the following data on the [Selection](#) sheet:



New Report Start Date [UTC]	21.06.2015
New Report Start Time [UTC]	09:40:50
Period Start Date	01.06.2015
Period End Date	30.06.2015
First Run in Period	No

Note that *First Run in Period* is set to *No* this time.

## 4.4.2.4 How Effective Period Selection Works

You can use filter parameters on the *Selection* sheet to define effective period selection, thus selecting employees that fulfill certain conditions in a specific time frame.

The parameters you define on the *Selection* sheet of the Employee Delta Export for Microsoft Excel must apply to Job Information or Compensation Information. The time frame is specified by the data entered in the selection fields *Select From Date* and *Select To Date* on the *Selection* sheet. By using both selection fields, you specify a period in time with a beginning and an ending. If you only use one selection field, you specify an open time period with either a fixed start date or a fixed end date.

### Example

Without effective period selection, you can only select employees that have been changed since the given *New Report Start Date* and that have worked, are currently working, or will be working at Company A.

#### Note

For the sake of simplicity, this example assumes that all employees have been changed since the *New Report Start Date*.

- Employee 1 worked at Company A from January 1 to March 30, 2016
- Employee 2 has been working at Company A since February 1, 2016
- Employee 3 will start working at Company A on January 5, 2016

Using effective period selection, you can limit the employees returned in the response by their jobs.

Selecting Employees by Their Jobs

Query	Response
Select all employees who worked at Company A in 2015	No employees are returned
Select all employees who worked at Company A in 2016	All employees are returned

Query	Response
Select all employees who worked at Company A between February 1 and April 1, 2016	Returns Employee 1 and Employee 2
Select all employees who have worked at Company A since June 1, 2016	Returns Employee 2 and Employee 3

### 4.4.3 Summary Sheets

The [Summary](#) sheets of Employee Delta Export for Microsoft Excel give a summary of the last extraction run.

The [Summary](#) worksheets contain the following information:

- The period for which changes were considered
- Filter values that were used for the extraction
- The number of employees that fulfilled the specified filter criteria

### 4.4.4 Systems Sheet

The Employee Delta Export for Microsoft Excel has a [Systems](#) sheet, which contains all the systems to which you can connect.

The Employee Delta Export Add-In for Microsoft Excel connects to an SAP SuccessFactors system using HTTPS and retrieves data by calling the `Compound Employee` API and other APIs. You may connect to different systems, for example, a test system and production system by choosing the desired system in the [Choose System](#) field on the [Selection](#) sheet.

The systems in the dropdown list are specified on the [Systems](#) sheet, which is typically hidden. Store your favorite systems on this sheet. The columns [Name](#), [URL](#), and [Company](#) are mandatory. When you select [Get Data](#), a logon dialog pops up and you have to enter the following OAuth 2.0 keys from the selected system:

- [User ID](#)
- [API Key](#)
- encrypted [Private Key](#) from the X.509 Certificate

Here's an example of what the [Systems](#) sheet looks like:

	A	B	C
2	Name	Url	Company
3	Production	<a href="https://api099.successfactors.eu">https://api099.successfactors.eu</a>	myprod
4	Test	<a href="https://api099.successfactors.eu">https://api099.successfactors.eu</a>	mytest
5			

Systems Sheet with Connecting Backend System

## Related Information

[Setting up the OAuth 2.0 Application \[page 11\]](#)

### 4.4.5 Config Sheet

The [Config](#) sheet of the Employee Delta Export for Microsoft Excel contains configuration parameters that control how the extraction is performed and how the results are displayed.

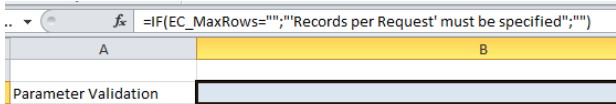
Some sections of the [Config](#) worksheet are locked to prevent unintended changes. If you want to change these sections anyway, you must unlock the sheet on the [Review](#) tab.

Some of the parameters are Microsoft Excel tables. Extending the tables must be done carefully. Make sure you don't enter data outside the table range. Otherwise, data may not be considered during processing.

Here's a list of the configuration parameters and information about how you use them.

#### Configuration Parameters

Parameter	Description
Show Unchanged Values	This parameter controls whether unchanged values in a changed segment are shown in the result sheets. As a standard, a value in a <a href="#">New</a> column (for example <a href="#">Address 1 [New]</a> ) is only then shown if it differs from the values in the corresponding <a href="#">Old</a> column or if it is explicitly configured to always be shown. However, when this parameter is set to <a href="#">True</a> , the <a href="#">New</a> / <a href="#">Old</a> pairs are always filled, if there are values in the backend.
Hire/Rehire Conditions	This field contains an XPath expression that is applied to the result of the Compound <code>Employee</code> API call in order to figure out whether an employee is rehired. This condition typically checks whether the event has changed in a certain way. If an employee is recognized as rehired, the full set of data is shown for this employee - not just the changed data.
File Name Pattern	This cell contains a formula that calculates the file name under which the workbook is saved when you use the <a href="#">Save Copy</a> button on the <a href="#">SuccessFactors</a> tab on the Microsoft Excel ribbon. You may include any of the selection parameters in the file name. In addition, the placeholders <a href="#">%path%</a> and <a href="#">%date%</a> may be used. <a href="#">%path%</a> is replaced by the folder in which the active workbook resides. <a href="#">%date%</a> is replaced by the current date/time in the format YYYY-MM-DD_HH-MM-SS.

Parameter	Description
Parameter Validation	<p>This cell contains a formula that checks whether the parameters in the <a href="#">Selection</a> sheet are consistent. The report will not start unless this formula returns blank. In the case of inconsistencies, the formula will return an error message that is presented to the user when extraction is started. For example, the sample workbook contain the formula shown in the graphic below, which checks if the <a href="#">Records per Request</a> parameter is filled. If not, the message '<a href="#">Records per Request</a>' must be specified is brought up and the report is aborted.</p>  <p>In a customer scenario, the formula may be used to enforce that one or more filter values are specified on the <a href="#">Selection</a> sheet.</p>
Update New Report Start Date	<p>This parameter controls whether the <a href="#">New Report Start Date/Time</a> fields on the <a href="#">Selection</a> sheet are updated automatically after an extraction run. The standard setting for this parameter is <a href="#">Yes</a>. This ensures that the user running the report doesn't need to worry about manually updating the extraction start date/time. The user simply needs to choose the <a href="#">Get Data</a> button to get the changes that have taken place in the system since the last extraction run.</p> <p>However, there are cases where it makes sense to set the parameter to <a href="#">No</a>. For example, when you want to test the extraction with the same base date/time over and over again.</p>
Records per request	<p>The number of records to be retrieved with one API call. If there are more records that meet the selection parameter, subsequent queryMore requests are sent to retrieve more records. The greater this parameter is, the less overhead is caused, however, the longer it takes to process one chunk of data.</p> <p><b>Recommendation</b></p> <p>200 records for each request.</p>
QueryMore Iterations	<p>Maximum number of queryMore calls.</p> <p>Set this parameter to <a href="#">Unlimited</a> in production environments.</p>
Use PaymentInformationV3	<p>Set this parameter to <a href="#">Yes</a> to enable Payment InformationV3 for the single-sheet workbook instead of Direct Deposit.</p>
Build Dynamic Request	<p>Set this parameter to <a href="#">Yes</a> to use the <a href="#">Requested Segments</a> table as the source of the query call.</p>

Parameter	Description
Table Format	<p>The area labeled <a href="#">Table Format</a> specifies the formatting which is used for the result sheets. Note that this part includes named cells that should not be deleted. Otherwise the rendering code will no longer run.</p> <p>To change the formatting of the table in the data sheets, change the formatting of this table accordingly using Microsoft Excel standard formatting options. Make sure you don't delete or add cells in this table.</p>
Format for Changed Fields	<p>The cell right to the label provides the formatting that is applied to changed fields in the result sheets. Only the values in the <a href="#">New</a> columns will be formatted like that.</p>
Additional Filters (ANDed)	<p>This table defines filter expressions (XPath) that are applied locally in Microsoft Excel to the response XML of the <code>Compound Employee</code> API to check whether an employee is included. An employee is in if they match all XPath expression in the list. For more information, refer to the <a href="#">Filter Logic</a> section.</p>
Additional Filters (ORed)	<p>This table defines filter expressions (XPath) that are applied in Microsoft Excel locally to the response XML of the <code>Compound Employee</code> API to check whether an employee is included. An employee is in if they match all conditions of the <a href="#">Additional Filters (ANDed)</a> table and they match any of the XPath expression in this list. For more information, refer to the <a href="#">Filter Logic</a> section.</p>
Keep Intermediate Results	<p>This parameter controls whether intermediate results are kept in the Microsoft Excel workbook beyond the end of the extraction run. The intermediate results (kept on hidden sheets of the workbook) are required for calculating the delta information and contain more data than are shown on visible sheets. Therefore, for security reasons, this option should be set to <a href="#">No</a> in productive environments. This ensures that intermediate results are deleted once the visible sheets are filled.</p> <p>You have to set this to <a href="#">Yes</a> when you want to send the filled report to SAP for troubleshooting purposes.</p>
Test Mode	<p>This parameter controls whether data is fetched from the system or whether the intermediate results contained in the workbook are used. This mode is typically used for troubleshooting when the person debugging the code has no access to the system. A prerequisite is that the intermediate results have been kept in the workbook by retrieving with <a href="#">Keep Intermediate Results = Yes</a> before.</p> <p>This parameter must be set to <a href="#">Off</a> in productive environments.</p>

Parameter	Description
Change Sheets	This table contains the names of the sheets that are intended to receive changed data in contrast to the table receiving new data for hire and rehires. If you don't need one of these sheets to be filled, you may hide or delete it and remove the corresponding entry from this list. In case you introduce a new sheet containing changes, you have to add the sheet name to this table. Otherwise it won't be filled with data. The multi-sheet workbook contains a <i>Payment Info</i> sheet that is not a part of this table. For more information refer to <a href="#">Using Payment Information in the Result Sheets [page 68]</a> .
Hire Sheet	This field contains the name of the sheet to receive new data for hired or rehired employees. You do not necessarily need such a sheet. For example, the workbook template CompEmp-Delta_multiSheet.xlsx has no <i>Hire Sheet</i> because new data goes into the entity sheets.
Requested Segments	<p>This table contains the technical segment names used to query data from the system using the Compound Employee API. All segments included in this table are supported by the Compound Employee API. Remove the segments from this table that you don't want to use for data processing in the Delta Export workbook.</p> <p>The result sheets for <i>Recurring Deduction</i>, <i>One Time Deduction</i>, <i>Personal Documents Information</i>, and <i>Alternative Cost Distribution</i> are only filled with data if their data segments are requested in this table. They are not part of the standard query. For more information, refer to <a href="#">Choosing Which Segments Are Extracted [page 69]</a>.</p>

## Related Information

[How Filtering Works \[page 36\]](#)

### 4.4.6 Errors Sheet

Employee Delta Export for Microsoft Excel has an *Errors* sheet, which shows all employees who could not be replicated for whatever reason.

Employees shown on the *Errors* worksheet will not appear on any of the results sheets. You need to take this very seriously and handle these employees manually.

The reasons are typically severe inconsistencies in the time slices of effective-dated entities for this employee. For example, if a compensation time slice starts at a time for which no job information time slices exists.

## 4.4.7 Warnings Sheet

The [Warnings](#) sheet of the Employee Delta Export for Microsoft Excel shows employees for which certain fields could not be retrieved due to picklist or foundation object configuration issues.

Employees shown on the [Warnings](#) worksheet also show up on the result sheets as usual. However, the fields indicated in the warning remain empty.

A very common example is following warning:

*"custom-string1" can't be returned. Please make sure that the external code for picklist "currency" with option ID "12722" is valid.*

This means that the field [custom-string1](#) on the entity that is shown with the warning cannot be filled for the employee in question since the data model has a picklist "currency" configured for this field which is not properly configured. The `Compound Employee` API, and as a result also the Microsoft Excel application, require the external code to be filled for all picklist options. Otherwise such a warning appears.

Another reason could be that the picklist assignment was changed in the data model, so that a picklist was assigned to a formed free-text field or the picklist assignment was changed to a different picklist. In this case, the valued assigned to employees based on the formed picklist is no longer valid for the new picklist.

## 4.4.8 Internal Sheets

A couple of worksheets in the reporting workbook of the Employee Delta Export for Microsoft Excel are used for internal purposes and are not shown to the end user unless you make them visible.

To make internal worksheets visible, use the [Show Internal Sheets](#) function on the [Developer](#) tab. You shouldn't change or remove these sheets.

Here's a short overview of the internal sheets:

### Lookup

This sheet contains a couple of tables with foundation data that are used to show the user-friendly descriptions of fields on the result sheets that are based on foundation objects. For example, to show the description of a pay component. This is sometimes desired because the `Compound Employee` API only returns the external codes of these fields. For more information on how to make use of the foundation data on this sheet, refer to the [Extending Employee Delta Export](#) section.

The data on this sheet is refreshed from the system automatically when an unresolved reference to lookup data is detected after filling the result sheets.

### LookupPerson

This sheet contains additional data for persons who are referred to on the result sheets. This is useful for cases where the `Compound Employee` API only provides the user ID of these persons. The job information, for example,

contains the field [Manager](#) which is actually the user ID of the manager. If you want to show the first and last name of this manager, you can make use of the [LookupPerson](#) sheet. For more information, refer to the *Extending Employee Delta Export* section.

## Picklists

This sheet contains all picklist entries for all picklists found in the system. This is useful when you want to show the user-friendly description of a field on the result sheets, which is based on a picklist. In the sample workbook `CompEmpDelta_multiSheet.xlsx`, the [Address Type](#) field on the [Address](#) sheet is such a case. This is sometimes desired since the `Compound Employee` API only returns the external codes of these fields. For more information on how to make use of the foundation data on this sheet, refer to the *Extending Employee Delta Export* section.

## Sheets DeltaResult, SnapshotResult, FullResult, Responses

These sheets are used for internal processing. During a report run, they receive the raw data returned by the API calls to the system. As a default, the content of the sheets is deleted at the end of a report run. For troubleshooting purposes, however, it makes sense to keep the data on these sheets. This can be controlled by the configuration parameter [Keep Intermediate Results](#).

## Scrambling

This sheet contains a table of XPath expressions that are used by the [Scramble Data](#) function on the [Developer](#) tab. These XPaths specify the fields that will be affected by the scrambling algorithm.

## Related Information

[Extending Employee Delta Export for Microsoft Excel \[page 57\]](#)

## 4.5 Tabs on the Ribbon

The Employee Delta Export Add-In for Microsoft Excel creates a [SuccessFactors](#) tab and adds a [SuccessFactors](#) group to the [Developer](#) tab in the Microsoft Excel ribbon.

[SuccessFactors Tab \[page 49\]](#)

Once you've installed the Employee Delta Export Add-In for Microsoft Excel, the [SuccessFactors](#) tab will appear in the ribbon.

[Developer Tab \[page 49\]](#)

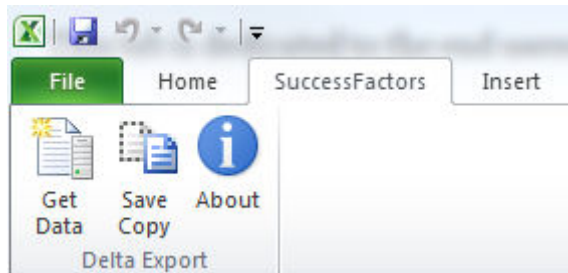


The Employee Delta Export Add-In for Microsoft Excel adds a *SuccessFactors* group to the *Developer* tab.

## 4.5.1 SuccessFactors Tab

Once you've installed the Employee Delta Export Add-In for Microsoft Excel, the *SuccessFactors* tab will appear in the ribbon.

The *SuccessFactors* tab is for the end users of the Employee Delta Export for Microsoft Excel.



SuccessFactors Tab

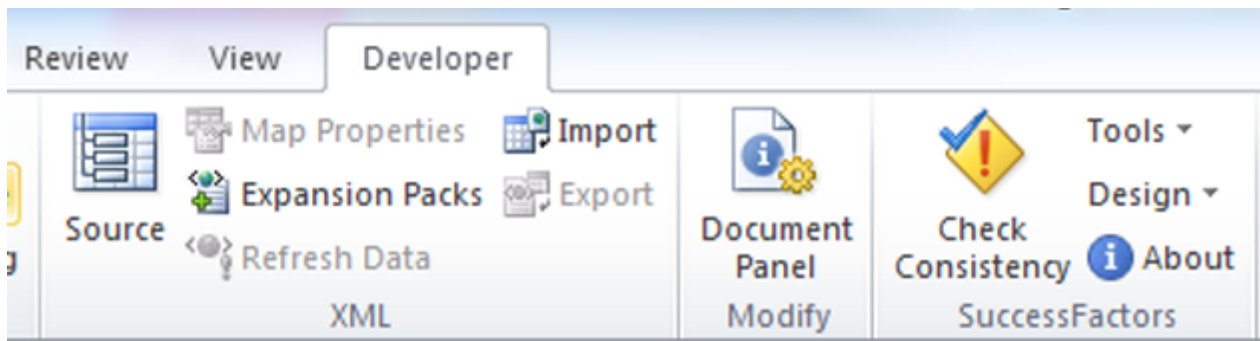
The following functions are available on the *SuccessFactors* tab:

- *Get Data*: Starts the extraction of data from the system and subsequent post-processing.
- *Save Copy*: Saves a copy of the Microsoft Excel that is intended to be sent out to other people in or outside your organization. This copy doesn't contain any internal sheets or hidden business data.
- *About*: Gives you the version information for the installed add-in.

## 4.5.2 Developer Tab

The Employee Delta Export Add-In for Microsoft Excel adds a *SuccessFactors* group to the *Developer* tab.

The functions in the *SuccessFactors* group are useful especially for an admin who needs to customize the sample workbooks.



### Note

The *Developer* tab is only visible if you have enabled it in Microsoft Excel.

#### [Consistency Check for the Workbook Configuration \[page 50\]](#)

The consistency check on the [Developer](#) tab of Employee Delta Export for Microsoft Excel verifies that the configuration is consistent.

#### [Tools on the Developer Tab \[page 50\]](#)

The Employee Delta Export for Microsoft Excel provides tools for making personal data anonymous, for showing or hiding internal sheets or configuration rows and columns, and for upgrading the workbook.

#### [How to Adjust the Workbook Design \[page 52\]](#)

Using the [Design](#) function on the [Developer](#) tab, you can add columns to or remove columns from the Employee Delta Export for Microsoft Excel.

## 4.5.2.1 Consistency Check for the Workbook Configuration

The consistency check on the [Developer](#) tab of Employee Delta Export for Microsoft Excel verifies that the configuration is consistent.

### Caution

Performing a consistency check will delete the data on the result sheets.

Use the consistency check after you've changed the configuration part of a workbook. In particular the following checks are performed:

- Are the XPath's contained in rows 1 and 2 of the Result sheets correct?  
If the rows contain only simple segment and field XPath's these are cross-checked against the signature of the Compound Employee API in the backend. If the XPath contains logical or other operators it is only checked if the XPath's are syntactically correct.
- Are the [New](#) / [Old](#) column pairs considered in the formula of the column named [EC\\_RelevantChanges](#)?  
This is essential because otherwise changes in these columns are not considered during delta calculation.
- Are segment XPath's maintained for columns that have correct lookup formulas?

## 4.5.2.2 Tools on the Developer Tab

The Employee Delta Export for Microsoft Excel provides tools for making personal data anonymous, for showing or hiding internal sheets or configuration rows and columns, and for upgrading the workbook.

The following tools are available on the [Developer](#) tab of the Employee Delta Export for Microsoft Excel:

- [Scramble Data](#)
- [Show/Hide Internal Sheets](#)
- [Show and Hide Configuration Rows/Columns](#)
- [Upgrade Workbook](#)

Here's more info about these tools:

## Scramble Data

The [Scramble Data](#) tool makes the personal data on the result and internal sheets anonymous. Use this before sending a spreadsheet filled with sensitive data to unauthorized persons.

The rules for this scrambling are on the hidden [Scrambling](#) sheet. The sample workbooks deliver a set of XPath paths appropriate for the standard Employee Central data models. If a customer's data model introduces additional personal data in custom fields, it's up to the customer to add rules for these fields as well.

### Note

Data on the hidden sheets [Delta Result](#), [Snapshot Result](#), and [Full Result](#) may contain data that is not shown on the visible result sheets. Rules will cover this data as well.

## Show/Hide Internal Sheets

Hides or unhides all sheets used for internal purposes. Show these for troubleshooting in case unexpected results are shown on the result sheets.

## Show and Hide Configuration Rows/Columns

Shows and hides all rows and columns with configuration information, these are the rows one, two, and three of the result sheets

Use this function rather than manually hiding the configuration rows and columns to ensure that all such rows and columns are affected.

## Upgrade Workbook

Use [Upgrade Workbook](#) to upgrade your customer's workbook with new features delivered by SAP SuccessFactors. The result is a workbook with customer-specific enhancements as well as new features, such as new result sheets, new selection parameters on the [Selection](#) and [Summary](#) sheets, or new configuration parameters on the [Config](#) sheet. The result sheets containing business data are typically not affected by upgrades.

To upgrade customer-specific workbooks, do the following:

1. Download the most recent version of the workbook provided by SAP SuccessFactors from the [Employee Delta Export UI](#).
2. Open the customer-specific workbook. This workbook can be of version 1502 or higher.
3. Choose the [Upgrade Workbook](#) tool from under [Tools](#) on the [Developer](#) tab.

4. When prompted, open the standard workbook you just downloaded in step 1.
5. Follow the instructions.
6. Check the [Upgrade Log](#) sheet for messages and required follow-up actions.

## Upgrading Result Sheets

Before adding a new result sheet, the upgrade process checks if a sheet with the same name already exists. If, for example, the [Payment Info](#) sheet already exists, when it is about to be copied to the target workbook, the upgrade step is skipped entirely and a message is written to the [Upgrade Log](#). The upgrade log is a tab that is created for each upgrade run and is named [Upgrade Log <starting version> to <target version>](#) for example [Upgrade Log 1505 to 1511](#). It contains all information about which upgrade steps have been performed and all follow-up actions needed after the upgrade. Action items are highlighted in yellow.

## Upgrading Selection and Summary sheets

If new parameters need to be added to the [Selection](#) or [Summary](#) sheets, the upgrade process checks if the sheets still contain the default content as delivered in the standard workbooks. If the content is unchanged, the parameters are added in at the same place as they are located in the standard workbook of a higher version. Customer-specific parameters, comments, and any changed cell content is recognized as being different from the default content. In this case the new parameters are added to a new location at the bottom of the sheet. Feel free to move the parameters from their new location to a spot that suits you best.

### Note

When moving rows to another position of the sheet, make sure to copy the named ranges to the new location as well. Otherwise, the workbook might not work correctly.

## Upgrading the Config Sheet

If new configuration parameters need to be added, the whole existing [Config](#) sheet is replaced and renamed to [Config Upgrade <Date>](#). The values are still available in the renamed sheet and can be copied over to the new sheet. You can delete the renamed sheet after the upgrade process.

## 4.5.2.3 How to Adjust the Workbook Design

Using the [Design](#) function on the [Developer](#) tab, you can add columns to or remove columns from the Employee Delta Export for Microsoft Excel.

## Add Columns

The [Add Columns](#) function starts a wizard that enables you to add one or more columns based on fields that are available for a specific segment in the Compound Employee API signature. The selection table has the following columns:

- The column [Field Name](#) displays the technical name of the field in the Compound Employee API and a checkbox that allows the selection of multiple fields in one segment.

### Note

The selection of multiple fields from various segments isn't supported

- The *Column Heading* column contains the field label found in the data model. If no field label has been maintained in the data model, the value is derived from the technical name.
- The columns *New/Old Columns* and *Always Show Value* display how the generated column will be configured.

When a row in the field list is selected, the *Column Properties* are shown below the list.

In the *Column Properties*, you can change the column heading and the generation properties. Selecting the *Generate New/Old Columns* option creates a new/old column pair for the selected fields that displays the current value and the value before the change. Furthermore, these columns are considered in the formula in the *EC\_Relevant Changes* column. If this option isn't selected, only a single column is created that shows the current value (if that particular field has changed since the last extraction). By selecting the *Always Show Value in Column* option, you generate a column that always shows the current value of a field, even if the value hasn't changed since last extraction. If neither *Always Show Value in Column* nor *Generate New/Old Columns* are selected, the new column only displays the values that have changed since the last extraction. For fields that are picklist-based, lookup columns are generated that show the picklist labels. The columns that contain picklist values are generated and then hidden.

## Payment Information Segments in Add Columns

If you have enabled Payment Information and configured the workbook to use it, the *Add Columns* wizard displays all available fields for the `payment_information` segment and its subsegments. The `payment_information` segment has a hierarchical structure:

- `PaymentInformationV3`
  - `PaymentInformationDetailV3`
    - `PaymentInformationDetailV3USA`
    - ....
    - `PaymentInformationDetailV3GER`

Most of the relevant Payment Information fields are on the `PaymentInformationV3Detail` segment node. Therefore, the wizard uses this node as the base node. The parent node is the `PaymentInformationV3`, fields on the root node begin with `../` for example `../createdBy`. Fields that start with segment names are the country/region-specific child nodes of the Detail node. For example, `PaymentInformationDetailV3JPN/createdBy`. Even if a field on such a country/region-specific segment has the same name as a field on another country/region segment, they're generated as separate columns on the *Result* sheet.

### Note

It's possible to create custom associations on the `payment_information` segment and its subnodes. However, the associations won't be considered by the *Add Columns* wizard. Only fields on standard segments of `payment_information` are included on the field list.

## Hierarchical MDF Segments in Add Columns

In the [Add Columns](#) wizard, hierarchical MDF-based segments that have a header segment with items underneath, like for example [Alternative Cost Distribution](#) or [Recurring Deductions](#), are handled the same way as [Payment Information](#). The item level is then used as segment XPath. Fields of the header segment are addressed with `../` in the same way as is done for [Payment Information](#). Custom associations aren't reflected in the [Add Columns](#) wizard.

## Refresh Field List Cache

The [Add Columns](#) wizard uses an internal cache to display all fields that are available for the supported segments. With the help of the cache, only one login is needed to call a `describeEx` and retrieve the field list, which is then displayed. In cases where changes have been made to the company's data model that aren't yet reflected in the [Add Columns](#) field list, you can trigger a refresh using this option. However, this requires another login to call the `describeEx`.

### [Adding Columns Using the Wizard \[page 54\]](#)

Use the wizard provided by the Employee Delta Export for Microsoft Excel to add columns.

### [Deleting Columns Using the Wizard \[page 56\]](#)

Use the wizard provided by the Employee Delta Export for Microsoft Excel to remove columns.

### 4.5.2.3.1 Adding Columns Using the Wizard

Use the wizard provided by the Employee Delta Export for Microsoft Excel to add columns.

## Context

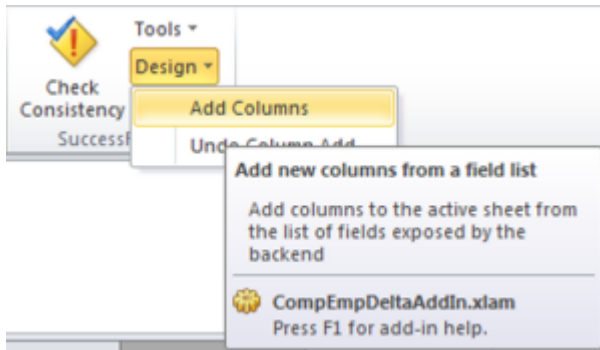
The [Add Columns](#) function starts a wizard that enables you to add one or more columns based on fields that are available for a specific segment in the `Compound Employee` API signature.

### → Remember

Make sure that a system is defined on the [Selection](#) sheet.

## Procedure

1. Go to the [Developer](#) tab and choose [Design](#) > [Add Columns](#).
2. Place your cursor in the column to the left of where you would like to add the new column and select [Add Columns](#).



#### Add Columns

3. Verify that the pre-selected segment name is the right one. If not, just select the desired segment from the list. This is the segment in the API signature that will be bound to the column.

Field Name	Column Heading	New / Old Columns	Always Show Value
<input type="checkbox"/> address1	Address1	True	False
<input type="checkbox"/> address10	Address10	True	False
<input type="checkbox"/> address11	Address11	True	False
<input type="checkbox"/> address12	Address12	True	False
<input type="checkbox"/> address13	Address13	True	False
<input type="checkbox"/> address14	Address14	True	False
<input type="checkbox"/> address15	Address15	True	False
<input type="checkbox"/> address16	Address16	True	False
<input type="checkbox"/> address17	Address17	True	False
<input type="checkbox"/> address18	Address18	True	False
<input type="checkbox"/> address19	Address19	True	False
<input type="checkbox"/> address1_alt1	Alt1 address1	True	False
<input type="checkbox"/> address1_alt2	Address1 Alt2	True	False

Column Properties

Field Name:

Column Heading:

☐ Generate New / Old Columns    ☐ Always Show Value In Column

Cancel    Add Columns

#### Segment Selection

4. Select [Add Columns](#) to add the selected columns to the currently selected sheet.

#### Note

If you use the [Add Columns](#) function on the [Hire](#) sheet of a single-sheet type workbook, the options [Generate New/Old Columns](#) and [Always Show Value in Column](#) are not available.

## 4.5.2.3.2 Deleting Columns Using the Wizard

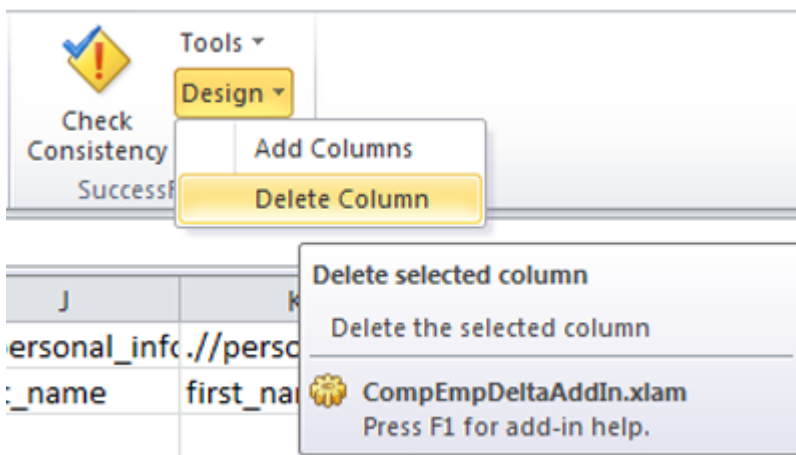
Use the wizard provided by the Employee Delta Export for Microsoft Excel to remove columns.

### Context

The [Delete Column](#) function starts a wizard that enables you to delete a selected column from a result sheet. The difference to the standard Microsoft Excel delete function is that this function preserves the consistency of the workbook from an SAP SuccessFactors perspective. You can only delete single columns at a time.

### Procedure

1. Go to the [Developer](#) tab and choose ► [Design](#) ► [Delete Column](#) ►.



Delete Column

2. If the column to be deleted is used in the change formula of the active sheet, the formula is adjusted automatically.
3. If the column you have selected for deletion is the last column referenced in the [EC\\_RelevantChanges](#) formula, the deletion is not executed and an error message is displayed. This ensures the functional correctness of the [EC\\_RelevantChanges](#) formula.



# 5 Extending Employee Delta Export for Microsoft Excel

You can use the default reporting workbooks provided by the Employee Delta Export as a starting point, but you need to customize them to fulfill the requirements of a productive use case.

The Employee Delta Export comes with two different variants of reporting workbooks. Customizing can just be changing the configuration data that is included in the reporting templates. No programming is required for this. Some XML knowledge is beneficial, however, since the configuration uses XPath expressions to identify the elements in the XML response of the `CompoundEmployee` API calls.

Customizations that require a change in the add-in should be sent to SAP SuccessFactors as a requirement.

Changes in the add-in's VBA code are not recommended since these will be overwritten by the next update.

## [What Types of Changes Can Be Applied to Reporting Workbooks? \[page 58\]](#)

When extending Employee Delta Export for Microsoft Excel you can apply three categories of changes to the reporting workbooks.

## [How Column Configuration Works \[page 59\]](#)

Before you start adding new columns to a result sheet you need to understand how column configuration works in the Employee Delta Export Add-In for Microsoft Excel .

## [Adding New Columns from the Database \[page 60\]](#)

Add new columns from the database to extend the result sheets of the Employee Delta Export for Microsoft Excel.

## [Adding Calculated Columns \[page 61\]](#)

Extend the result sheets of Employee Delta Export for Microsoft Excel by adding a column that contains a calculated value.

## [Adding User-Friendly Descriptions for Picklist-Based Columns \[page 62\]](#)

Display your own user-friendly descriptions in Employee Delta Export for Microsoft Excel, rather than the external codes for fields based on picklists.

## [Adding User-Friendly Descriptions for Foundation Object-Based Columns \[page 63\]](#)

Display your own user-friendly descriptions rather than the external codes for fields based on foundation objects (FOs).

## [Deleting Columns \[page 64\]](#)

Delete columns from Employee Delta Export for Microsoft Excel if you don't need them.

## [Adding Personal Data to User ID Fields \[page 65\]](#)

Show personal information like first name or last name along with the user ID in Employee Delta Export for Microsoft Excel.

## [Adding Information About Workflow Approvals \[page 66\]](#)

Show information about workflow approvals related to a segment in the Microsoft Excel that's populated by the Employee Delta Export Add-In.

## [Adding New Filter Parameters \[page 67\]](#)

Add more filters to the [Selection](#) sheet for Employee Delta Export for Microsoft Excel. It already has multiple default fields you can use for filtering employees, but you can add more if necessary.

### [Using Payment Information in the Result Sheets \[page 68\]](#)

If you have enabled Payment Information in Provisioning, there are a few steps you need to take to make the *Payment Info* sheet visible in the multi sheet workbook as well as new columns on the *Hire* and *Change* sheets of the single sheet workbook.

### [Choosing Which Segments Are Extracted \[page 69\]](#)

Use the *Requested Segments* table in Employee Delta Export for Microsoft Excel to choose which segments are extracted from the backend, thus reducing the amount of data that is being retrieved.

### [Dos and Don'ts for Using the Employee Delta Export Add-In for Microsoft Excel \[page 70\]](#)

Check out some changes that you can make when adjusting the Employee Delta Export Add-In. And see what we don't recommend doing.

## 5.1 What Types of Changes Can Be Applied to Reporting Workbooks?

When extending Employee Delta Export for Microsoft Excel you can apply three categories of changes to the reporting workbooks.

These are the types of changes you can make:

- Simple layout changes:
  - Change column headers
  - Delete columns
  - Delete an entity sheet: delete the entire worksheet and remove it from the configuration on the *Config* sheet
  - Add new calculated columns
  - Remove filter parameters: delete the row on the *Selection* sheet and ensure it is not used on the *Config* sheet
  - Change the file name pattern: see section *Config Sheet*.
  - Add new rows between row 3 and row 5 of a result sheet.
- More sophisticated changes (requires some knowledge of XML and the Compound Employee API):
  - Add new columns from the database
  - Add user-friendly descriptions for picklist-based columns
  - Add user-friendly descriptions for FO-based columns
  - Add personal data for user ID fields
  - Add information about workflow approvals
  - Add filter parameters
  - Add validation rules for filter parameters
- Destructive changes (will prevent the extraction from running properly):
  - Deletion of one of the non-entity related sheets such as *Summary*, *Selection*, *Config*, and so on
  - Delete one of the first three (hidden) rows of the entity sheets
  - Change a name assigned to individual cells, rows, or columns
  - Add text adjacent to tables in the *Config* sheet (for example additional explanations)
  - Add rows after the header row (originally row 5) of the entity sheet

## 5.2 How Column Configuration Works

Before you start adding new columns to a result sheet you need to understand how column configuration works in the Employee Delta Export Add-In for Microsoft Excel .

Column configuration is in the hidden rows one, two and three of each result worksheet. You can make configuration visible using the function [Show Configuration Rows/Columns](#) in the [SuccessFactors](#) group on the [Developer](#) tab.

**Row 1** contains an XPath expression identifying the segment in the XML result of the Compound Employee API that carries the field to be shown in the column in question. For example, the XPath `./personal_information` identifies the **bold** element in the XML result below.

```
<person xmlns="urn:sfobject.sfapi.successfactors.com">
  <birth_name>Kendrick</birth_name>
  <personal_information>
    ...
    <first_name>Amy</first_name>
    ...
    <last_name>Kendrick</last_name>
    ...
    <start_date>2014-07-11</start_date>
  </personal_information>
  ...
</person>
```

**Row 2** contains an XPath fragment that identifies the element starting from the corresponding segment XPath in row 1. For example, if you want to show the field `first_name` in the column just enter `first_name` in row 2.

**Row 3** can contain a valid Microsoft Excel formula that calculates the values in this column from other columns. The formula must use cells references in row 3 to refer to values in the same row. The formula will be copied to rows below automatically.

### Note

Cells containing formulas must have the cell format [General](#). Make sure that the cell format is correct before you enter a formula.

The example below shows the configuration for first name, last name, and formatted name calculated from the first and last name. Note especially the formula in cell G3 of the graphic below.

	D	E	F	G	
	<code>./personal_information</code>	<code>./personal_information</code>	<code>./personal_information</code>		
	salutation	last_name	first_name		date
				<code>=F3 &amp; " " &amp; E3</code>	
	Personal Information				
	Title	Last Name	First Name	Formatted Name	

Configuration for First Name, Last Name, and Formatted Name

## 📌 Note

All columns of the same segment XPath should be configured contiguously. A column with a different segment XPath B in between a sequence of columns with XPath A leads to unexpected results. However, calculated columns interrupting the sequence is not harmful.

## 5.3 Adding New Columns from the Database

Add new columns from the database to extend the result sheets of the Employee Delta Export for Microsoft Excel.

### Context

For the following description let's assume you want to add the field *custom\_string5* of Personal Information.

### Procedure

1. Figure out the field name of the field in the signature of the `Compound Employee API`.

For a reference of valid fields please go to the *SFAPI Data Dictionary* in *Admin Tools*. The dictionary says the field name is */person/personal\_information/custom\_string5*

2. Make the first three rows of the worksheet visible. These rows contain configuration information.
3. Insert a new column where you want the additional data to appear.
4. In the first row of the new column, add `./person_information`, which is an XPath expression selecting the time slices of personal information within the result of the `Compound Employee API`.

The starting point of the XPath is assumed to be the `<person>` element. XPath expressions may also include restrictions like `./address_information[address_type='home']`. The XPath can also make use of content in neighboring columns of the same row. In order to do this, introduce a name for the column with the prefix `EC_Col_` and use the part following the prefix in the XPath.

### 🔗 Example

The XPath `./address_information[start_date>='%StartDate%']` refers to a column with the name *EC\_Col\_StartDate*.

5. In the second row of the new column add *custom\_string5*, which is the field name of the new field in the result of `Compound Employee API`.

The final XPath picking the field value out of the XML result is concatenated from the segment XPath above and the field name, which is in fact taken as an XPath fragment.

6. Add the newly added columns to the formula that determines if a row is relevant.

The purpose of the Employee Delta Export application is to show only records that shall be forwarded to a payroll system or a similar system. Typically, these are the records which are new or have changed significantly

since a certain point in time. For this purpose each sheet contains a column with a formula that determines if a row is relevant.

- a. Go to the column with the name *EC\_RelevantChanges* on the same sheet and adapt the formula in row 3 of this column.

The formula will return a value unequal blank if the row contains relevant changes. This ensures that the row appears on the result sheet. Note that this column may be hidden. You can figure out where it is by using the *Name Manager* function on the *Formulas* tab in Excel.

The next time you run the extraction, the new column will be populated if the field value has changed in Employee Central since the specified report start date.

- b. If you want to see both the current field value and the old value on the sheet, introduce a second column for the same field.

This works mostly as described above with the difference that the field name entered in the second row must be *custom\_string5/@previousValue*. This denotes the value as it was at the point in time of the last extraction.

- c. If you want to see the current value regardless of whether it was changed since the last extraction or not, use *custom\_string5/@currentValue* as a field name in the second row of the column in question.

This makes sense for key fields as well as for attributes such as currency.

- d. In the segment XPath, you can use a notation such as `full://personal_information`.

This denotes that the column is filled from the full extract rather than from the delta extract. This ensures that the field is filled even if *personal\_information* (in this example) has not actually changed.

#### Note

Using the *full://* notation in the field *XPath row* makes it easier to sort the columns.

The columns *User ID* and *Person ID* (columns B and C) are always sortable even though they don't use the *full://* notation.

- e. You may populate the columns of a particular sheet from different segments in the API result structure.

Note, however, that data from different segments may go into different rows in the worksheet. This may lead to unexpected results.

## 5.4 Adding Calculated Columns

Extend the result sheets of Employee Delta Export for Microsoft Excel by adding a column that contains a calculated value.

### Procedure

To add a column that contains a value calculated from other columns of the same row, enter a formula into row 3 of the column that refers to other cells in the same row.

### Note

Cells containing formulas must have the cell format *General*. Make sure that the cell format is correct before you enter a formula.

## 5.5 Adding User-Friendly Descriptions for Picklist-Based Columns

Display your own user-friendly descriptions in Employee Delta Export for Microsoft Excel, rather than the external codes for fields based on picklists.

### Context

The `CompoundEmployee` API only provides external codes for fields based on picklists. Here's how to display your own user-friendly descriptions rather than the external codes:

### Procedure

Add a new column with a formula that does a lookup to the *Picklists* sheet, which is typically hidden.

Take column *Address Type* in the sample reporting workbook `CompEmpDelta_multiSheet.xlsx` as a reference. It uses this formula:

```
=VLOOKUP(CONCATENATE("addressType|";G3);Picklists!A:D;4;FALSE)
```

The formula looks for the row containing the address type's external code on sheet *Picklists* and returns the value in column 4 which is in fact the label.

G	I
./address_information	
address_type/@currentValue	
	=VLOOKUP(CONCATENATE("addressType ";G3);Picklists!A:D;4;FALSE)
Address Type Code	Address Type
[New]	[New]

Lookup of Address Type Field

### Note

Cells containing formulas must have the cell format *General*. Make sure that the cell format is correct before you enter a formula.

## 5.6 Adding User-Friendly Descriptions for Foundation Object-Based Columns

Display your own user-friendly descriptions rather than the external codes for fields based on foundation objects (FOs).

### Context

The `CompoundEmployee` API only provides external codes for fields based on foundation objects. Here's how to display your own user-friendly descriptions rather than the external codes for FO-based columns.

#### Note

Time dependencies of foundation objects are not considered. Only the currently effective values are used.

### Procedure

1. Add a new column containing a formula that does a lookup to the [Lookup](#) sheet, which is typically hidden.

Use column [Pay Component](#) in the sample reporting workbook `CompEmpDelta_multiSheet.xlsx` as a reference. It uses the following formula:

```
=VLOOKUP(E3;EC_Lookup_PayComponent;2;FALSE)
```

E	F	
./paycompensation_recurring		./p
pay_component/@currentValue		star
	=VLOOKUP(E3;EC_Lookup_PayComponent;2;FALSE)	
Pay Component Code	Pay Component	

Lookup of Pay Component

This formula looks for the row containing the pay component's external code in table [EC\\_Lookup\\_PayComponent](#) on sheet [Lookup](#) and returns the value in column 4 which is in fact the label. The table is filled automatically when unresolved references are found after processing the result of the `CompoundEmployee` API call.

#### Note

Cells containing formulas must have the cell format [General](#). Make sure that the cell format is correct before you enter a formula.

2. Make sure the formula in the *EC\_RelevantChanges* column contains the source column, not the lookup column. This is to keep delta reporting consistent.

## 5.7 Deleting Columns

Delete columns from Employee Delta Export for Microsoft Excel if you don't need them.

### Procedure

1. Use  *Design* > *Delete Column*  on the *Developer* tab to remove columns from result sheets.

When deleting columns using this tool, all references stay valid.

2. Check the formulas in row 3 for *REF!* errors and resolve them, especially the *EC\_RelevantChanges* column. If the column you want to delete is referenced in a lookup as source column, you have to delete the lookup column as well. If the *REF!* error appears in the formula in the *EC\_RelevantChanges* column, delete the expression containing the *REF!* entry from the formula.

#### ❖ Example

If the formula in *EC\_RelevantChanges* looks like this:

```
=IF(OR(#REF!<>G3;H3<>I3);"X";"")
```

Change it to:

```
=IF(OR(H3<>I3);"X";"")
```

In this way, you make sure that the changed calculation still works as desired, because corrupt formulas will prevent the formula to work at all.

3. To verify that workbook configuration is still intact, use the *Check Consistency* function on the *Developer* tab after removing columns.

### Related Information

[How to Adjust the Workbook Design \[page 52\]](#)

[Consistency Check for the Workbook Configuration \[page 50\]](#)



## 5.8 Adding Personal Data to User ID Fields

Show personal information like first name or last name along with the user ID in Employee Delta Export for Microsoft Excel.

### Context

The `CompoundEmployee` API only returns an employee's user ID. Here's how to show personal information along with this ID:

#### Note

Only personal data for active users can be retrieved. The lookup logic will have an impact on performance.

### Procedure

Introduce a new column with a lookup formula like this:

```
=VLOOKUP(A3; EC_Lookup_PersonalInfo; 3; FALSE)
```

This formula refers to the table `EC_Lookup_PersonalInfo` on the `LookupPerson` sheet. Once the formula is detected by the delta reporting logic, the personal information is retrieved for the user IDs found in the column referred to by the formula (in this case column A).

	A	B	
1			
2	person_id_external		
3		=VLOOKUP(A3; EC_Lookup_PersonalInfo; 3; FALSE)	=VLOOKUP(
6	User Id	Last Name	First Name
7	kkkk	=VLOOKUP(A7;EC_Lookup_PersonalInfo;3;FALSE)	=VLOOKUP(

Lookup of Last Name

## 5.9 Adding Information About Workflow Approvals

Show information about workflow approvals related to a segment in the Microsoft Excel that's populated by the Employee Delta Export Add-In.

### Procedure

1. Introduce a column that is mapped to the field `workflow_request_id` of the segment in question.

This is supported for the following segments of the `CompoundEmployee` API:

- `personal_information`
- `job_information`
- `compensation_information`
- `pay_compensation_non_recurring`

2. Introduce a column with a lookup formula like this:

```
=VLOOKUP(J3; EC_Lookup_Workflow; 4; FALSE)
```

This formula refers to the table `EC_Lookup_Workflow` on sheet `LookupOData`. Once the formula is detected by the Employee Delta Export logic, the workflow information is retrieved for the workflow requests found in the column referred to by the formula (in this case column J).

J	K	L
./personal_information		./personal_information
workflow_request_id		
	=VLOOKUP(J3;LookupOData!A:E;4;FALSE)	=VLOOKUP(J3;LookupOData!A:E;5;FALSE)
WF Req Id	WF Last Name	WF First Name
	=VLOOKUP(J6;LookupOData!A:E;4;FALSE)	=VLOOKUP(J6;LookupOData!A:E;5;FALSE)
803	=VLOOKUP(I7;LookupOData!A:E;4;FALSE)	=VLOOKUP(I7;LookupOData!A:E;5;FALSE)

Lookup of First Name and Last Name of Approver

#### Note

This logic will have an impact on performance.

## 5.10 Adding New Filter Parameters

Add more filters to the [Selection](#) sheet for Employee Delta Export for Microsoft Excel. It already has multiple default fields you can use for filtering employees, but you can add more if necessary.

### Context

There are two types of filters.

- Filters supported by the `CompoundEmployee` API in the backend
- Locally applied filters

Here's how you add these two types of filter parameters:

### Procedure

1. Add a new filter that is supported by the `CompoundEmployee` API.
  - a. Go to the [Selection](#) sheet and add a new row similar to the one for cost center.
  - b. For the cell where the filter value is entered, you need to introduce a name following the pattern `EC_Filter_<parameter name>` where `<parameter name>` is the name of the filter parameter defined in the `Compound Employee` API signature.  
  
For a reference of valid fields, go to the [SFAPI Data Dictionary](#) in the [Admin Tools](#). Uppercase/lowercase don't matter for the parameter name.
2. Add a new locally applied filter.
  - a. Go to the [Selection](#) sheet and add a new row similar to the one for country/region.
  - b. For the cell where the filter value is entered you need to introduce a name following the pattern `EC_FilterAtClient_<parameter name>` where `<parameter name>` is a name of your choice.
  - c. Go to the [Config](#) sheet and add the new parameter to the XPaths of one of the [Additional Filters](#) sections that are already in place or add a new XPath to one of these sections.

### Next Steps

You can also add filter parameters that are applied locally on the client after the result of the `CompoundEmployee` API call is available. To do so, add a new filter condition on the [Config](#) sheet that uses the new filter parameter as a placeholder.

## Related Information

[How Filtering Works \[page 36\]](#)

## 5.11 Using Payment Information in the Result Sheets

If you have enabled Payment Information in Provisioning, there are a few steps you need to take to make the [Payment Info](#) sheet visible in the multi sheet workbook as well as new columns on the [Hire](#) and [Change](#) sheets of the single sheet workbook.

### Procedure

1. To enable Payment Information in the multi-sheet workbook, make the following adjustments in the [Change](#) sheets table to fill the [Payment Info](#) result sheet:
  - a. Delete [Direct Deposit](#).
  - b. Add [Payment Info](#).

The [Payment Info](#) sheet is not used by the multi-sheet workbook by default, therefore, it is not included in the [Change Sheets](#) table on the [Config](#) sheet by default.

You can add either [Direct Deposit](#) or [Payment Info](#). If you include both sheets in the table, you'll get a warning. The request that is sent to the Compound Employee API changes depending on which sheet is added to the table.

If you have already enabled Payment Information in Provisioning but the workbook is configured to use [Direct Deposit](#), you will get a warning message and the call will not be executed. Same applies to the case where Direct Deposit is enabled in Provisioning and the workbook is configured to use [Payment Info](#).

2. To enable Payment Information in the single-sheet workbook, go to the [Config](#) sheet and set the [Select Option](#) for [Use Payment Information V3](#) to **Yes**.

Select Options	
Update New Report Start Date	Yes
Records per Request	100
QueryMore Iterations	unlimited
Use PaymentInformationV3	Yes

Select Option for Use Payment Information V3

If you have already enabled Payment Information in Provisioning, but the workbook is configured to use Direct Deposit, you will get a warning message and the call will not be executed. The same applies to the case where Direct Deposit is enabled in Provisioning and the workbook is configured to use Payment Information.

## 5.12 Choosing Which Segments Are Extracted

Use the [Requested Segments](#) table in Employee Delta Export for Microsoft Excel to choose which segments are extracted from the backend, thus reducing the amount of data that is being retrieved.

### Procedure

1. Go to the [Requested Segments](#) table on the [Config](#) worksheet.

By default, the [Requested Segments](#) table contains all segments supported by the Employee Delta Export Add-In by their technical name.

2. Remove the segments that aren't needed for data processing in your case, to reduce the amount of data that is extracted from the backend.
3. If necessary, add one or more of the following segments that aren't available in the standard query:
  - Deduction Recurring
  - One Time Deduction
  - Alternative Cost Distribution
  - Personal Documents Information

#### Note

Segments not listed in the Requested Segments are not supported by the Employee Delta Export Add-In.

By default, these result sheets are included in the [Change Sheet](#) table on the [Config Sheet](#), but aren't filled with results. Data extraction for these sheets will only happen if defined in the [Requested Segments Table](#).

#### Restriction

Recurring Deduction, One Time Deduction, and Alternative Cost Distribution must have start dates later than **October 10, 2015**. Delta calculation can't be performed for time periods before, due to missing MDF audit history data. You can't use the period-based delta functionality if you request One Time Deductions (deductions\_non\_recurring).

4. After adjusting the segment list in the [Requested Segments](#) table remember to set the configuration parameter [Build Dynamic Request](#) on the [Config Sheet](#) to [Yes](#) so that the request XML is built using the segments you maintained in the table.

### Related Information

[Config Sheet \[page 43\]](#)

[How Period-Based Delta Works \[page 40\]](#)

## 5.13 Dos and Don'ts for Using the Employee Delta Export Add-In for Microsoft Excel

Check out some changes that you can make when adjusting the Employee Delta Export Add-In. And see what we don't recommend doing.

### Field Sequence

Fields related to a particular segment in the result of the `CompoundEmployee` API must be in sequence without being interrupted by fields from other segments. The reason is that in sheets showing changes, a new row is started as soon as the segment changes. If the field sequence is interrupted, the fields of the same segment spread across different rows.

### Hidden Sheets and Rows

Don't delete hidden worksheets.

Don't delete the first 3 hidden rows of the worksheets that show the changes of an entity.

### Named Columns

Don't delete named columns on the change sheets. To figure out whether a column is a named column, select the column and check the name field in the area above the [A1](#) cell. Other columns can be deleted without restrictions, for example if the users no longer want to see it.

### Copying Sheets

You can copy sheets from one workbook to another to save time. A typical use case is to copy the [Config](#) or the [Systems](#) sheet to duplicate settings. To do so, right-click on the source sheet, choose [Copy](#), and choose the target. This can, however, create undesired copies of names that carry references to the source file. Use the [Name Manager](#) in Microsoft Excel to check your sheets and to remove the undesired names.

### Report Start Date

Make sure to set a [Report Start Date](#) and other available selection criteria, such as the [Requested Segments](#) table, to narrow down the target population that is extracted to the Delta Export workbook. Making these settings improves the performance and memory usage.

## Requested Segments Table

By default, the [Requested Segments](#) table is delivered containing all segments that are supported by the Compound Employee API. Remove all the segments of the table that you don't need for data processing. In that way, data extraction takes less time and post-processing has less data that needs to be managed.

### Note

Removing segments is only relevant if you've set the [Build Dynamic Request](#) parameter.

## Using Formulas

If the basic fields retrieved from the backend don't fully meet your requirements, you can introduce new columns that use formulas to calculate the content from other basic fields. For example, you can use a formula to compute a full address from the columns [postal code](#), [city](#), [country](#), and [address1](#). Enter the formula in row 3 of a change sheet. Only this ensures that it's copied to the data when the sheet is populated.

Columns containing formulas should typically not be considered in the formula contained in the column named [EC\\_RelevantChanges](#) if the original field is already in there. The reason is that the calculation can fail in a row for some reason and the row can appear unexpectedly.

## Population Changes

Replication of employee changes is easy when you have one payroll (or other type of) subsequent system to where you transfer the data. Things become more complex once you have various subsequent systems with different processors using different variants for the reporting workbooks. This is what we call populations. One population is a set of filter criteria, which specifies the employees going into one subsequent system.

Things become even more complex once employees move around between different populations. In this case, you need to ensure that processors recognize when an employee enters their population and only receive changes for employees who belong to their population.

## 6 Troubleshooting Employee Delta Export to Microsoft Excel

What you should do when an error occurs in Employee Delta Export for Microsoft Excel.

### Context

When an error occurs during data extraction that needs to be analyzed by your administrator or by SAP, proceed as follows:

### Procedure

1. Switch *Keep Intermediate Results* to *Yes* on the *Config* sheet.
2. Perform the extraction.
3. Scramble the data using the *Scramble Data* tool on the *Developer* tab to make personal data anonymous.
4. Save the workbook and send it to the person who analyzes the issue. Remember that the workbook contains all business-related data of the extracted employees.

### Related Information

[Config Sheet \[page 43\]](#)

[Tools on the Developer Tab \[page 50\]](#)



# Change History

Learn about changes to the documentation for Employee Delta Export Add-In for Microsoft Excel in recent releases.

## 1H 2024

Type of Change	Description	More Info
None	We did not update this document.	

## 2H 2023



Type of Change	Description	More Info
Changed	We have moved the Change History to the end of the guide.	<a href="#">Before You Start [page 4]</a>
Changed	We updated the supported versions for the Employee Delta Export Add-In for Microsoft Excel.	<a href="#">Technical Prerequisites for Using the Employee Delta Export for Microsoft Excel [page 7]</a>

# Important Disclaimers and Legal Information

## Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

## Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

## Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

## Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

## Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.



© 2024 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.